

# 1. Лабораторная работа №1

## Обобщения (Generics)

**Цель работы:** знакомство с обобщенными типами в языке C#.

**Задача работы:** научиться создавать обобщенные интерфейсы и классы, использовать обобщенные типы в приложениях .Net

**Время выполнения работы:** 4 часа (2 занятия)

**Результат выполнения работы:** программа, обеспечивающая функционал согласно заданию.

### 1.1.Задание для всех вариантов

- a) Создайте проект типа «Консольное приложение». Название проекта: GGGGGG\_NNN\_Lab5, где GGGGGG - номер группы, NNN – фамилия.
- b) Добавьте в проет папки Interfaces, Collections, Entities.
- c) В папке Interfaces опишите обобщенный интерфейс `ICustomCollection<T>`, декларирующий следующие методы и свойства:
  - `T this[int index]{ get;set; }` – индексатор коллекции
  - `void Reset()` – метод, устанавливает курсор в начало коллекции
  - `void Next()` – метод, перемещает курсор на следующий элемент коллекции
  - `T Current()` – метод, возвращает элемент текущего положения курсора
  - `int Count { get; }` – свойство, возвращает количество элементов в коллекции
  - `void Add(T item)` – метод, добавляет объект item в конец коллекции
  - `void Remove(T item)` – метод, удаляет объект item из коллекции
  - `T RemoveCurrent()` – метод, удаляет элемент текущего положения курсора
- d) В папке Collections опишите класс **MyCustomCollection<T>**, реализующий интерфейс **ICustomCollection<T>**. Для хранения объектов внутри класса **НЕ использовать** стандартные коллекции или массивы. Для возможности обхода коллекции каждый объект

коллекции должен иметь ссылку на следующий объект (либо на NULL, если это последний элемент коллекции)

- e) В папке **Contracts** создайте интерфейс, описывающий функции системы согласно индивидуальному заданию
- f) В папке **Entities** опишите классы сущностей согласно предметной области индивидуального задания. Один из классов должен реализовать интерфейс из п.5. Для хранения списков объектов используйте коллекцию, созданную в п 4
- g) В классе **Program** создать объекты и выполнить доступные функции программы согласно индивидуальному заданию. Если функция возвращает результат (например, общая стоимость), то этот результат вывести в консоль. Предусмотреть вывод в консоль наименований объектов, хранящихся в коллекциях.

**Примечание 1:** для заполнения списков исходными данными и для проверки функционала системы не обязательно делать интерактивное меню. Можно все действия «жестко» записать в коде класса Program.

**Примечание 2:** вывод информации в консоль должен осуществляться **Вне** разрабатываемой системы

**ОБЯЗАТЕЛЬНО:** при вычислении суммы, общей стоимости и т.д. использовать интерфейсы пространства имен GenericMath.

## 1.2. Индивидуальные задания

### 1. Предметная область: АТС.

На АТС хранится информация о всех клиентах станции. АТС имеет список тарифов на междугородние разговоры (например, стоимость одной минуты разговора в зависимости от города). Клиент АТС может совершать множество звонков в различные города.

Система должна:

- вводить информацию о тарифах;
- вводить информацию о клиентах
- регистрировать звонки;
- по введенной фамилии о клиенте определять стоимость всех сделанных им звонков в соответствии с действующими тарифами;

- вычислять общую стоимость всех выполненных на АТС звонков (использовать интерфейсы пространства имен GenericMath).
- вычислять количество всех звонков в заданный город

## **2. Предметная область: Вокзал.**

Касса вокзала имеет список тарифов на различные направления. При покупке билета регистрируется дата поездки. Пассажир покупает билеты на различные направления.

Система должна:

- позволять вводить данные о тарифах;
- позволять вводить паспортные данные пассажира
- регистрировать покупку билета;
- рассчитывать стоимость купленных пассажиром билетов (использовать интерфейсы пространства имен GenericMath);
- после ввода наименования направления, предоставлять список всех пассажиров, купивших на него билет;

## **3. Предметная область: ЖЭС.**

В ЖЭС хранятся тарифы на коммунальные услуги. ЖЭС имеет информацию обо всех жильцах. Каждый жилец может приобретать много услуг. При потреблении жильцами коммунальных услуг информация регистрируется в системе.

Система должна позволять выполнять следующие задачи:

- ввод тарифов
- ввод информации о жильцах и потребленных услугах;
- после ввода фамилии, выводить сумму всех потребленных услуг;
- вычислять стоимость всех оказанных услуг (использовать интерфейсы пространства имен GenericMath);
- вычислять общее количество заказов на заданную услугу

#### **4. Предметная область: Аэропорт.**

Касса аэропорта имеет список тарифов на различные направления. При покупке билета регистрируется дата полета. Пассажир может приобретать билеты на различные направления.

Система должна:

- позволять вводить данные о тарифах;
- позволять вводить паспортные данные пассажира
- регистрировать покупку билета;
- рассчитывать стоимость купленных пассажиром билетов (использовать интерфейсы пространства имен GenericMath);
- предоставлять список пассажиров на конкретную дату

#### **5. Предметная область: Банк.**

Информационная система банка хранит описание процентов по различным вкладам. При оформлении вклада регистрируется сумма вклада. Система хранит информацию о вкладчиках и сделанных ими вкладах. Каждый клиент может поместить в банк несколько вкладов.

Система должна позволять выполнять следующие задачи:

- хранить информацию о процентах по вкладам;
- хранить информацию о клиентах;
- пополнять клиенту величину вклада;
- вычислять общую сумму выплат по процентам для всех вкладов (использовать интерфейсы пространства имен GenericMath).
- вычислять проценты клиента по конкретному вкладу

#### **6. Предметная область: Отдел расчета зарплаты.**

Информационная система отдела расчета зарплаты на предприятии хранит данные о величине оплаты различных видов работ. Система хранит информацию о работниках предприятия. Работник может выполнять разные виды работ.

Система должна позволять выполнять следующие задачи:

- вводить информацию о различных видах работ;

- вводить информацию о работниках
- выводить информацию о работниках, выполнявших конкретную работу;
- после ввода фамилии выводить для работника зарплату (использовать интерфейсы пространства имен GenericMath);
- вычислять сумму выплат всем работникам;

### **7. Предметная область: Фирма грузоперевозок.**

Фирма имеет список тарифов по перевозке грузов в зависимости от направления и объема груза. Клиент регистрируется в системе, после чего может заказать перевозку определенного объема груза. Клиент может заказать несколько перевозок.

Система должна позволять выполнять следующие задачи:

- ввод тарифов;
- регистрация клиента и заказ на перевозку грузов;
- подсчет суммы заказа для определенного клиента;
- подсчет суммарной стоимости всех заказов (использовать интерфейсы пространства имен GenericMath).
- подсчет стоимости заказов на определенное направление

### **8. Предметная область: Гостиница.**

Информационная система гостиницы хранит информацию обо всех номерах и их стоимости. Система регистрирует клиентов. Каждый клиент может заказать несколько номеров. При попытке заказа номера, который занят, выводится предупреждение.

Система должна позволять выполнять следующие задачи:

- предоставление информации о номерах и их стоимости;
- регистрация клиента и заказ номера;
- вывод списка не занятых номеров;

- после ввода фамилии клиента, подсчет стоимости проживания (использовать интерфейсы пространства имен GenericMath).
- поиск самого популярного номера гостиницы (по количеству клиентов, заказывавших номер)

### **9. Предметная область: Интернет оператор.**

Провайдер имеет различные тарифы доступа в Интернет за 1Мбайт. Информационная система провайдера хранит данные о клиентах. Клиент может подписаться на несколько тарифов

Система должна позволять выполнять следующие задачи:

- ввод тарифов;
- регистрация пользователя;
- ввод данных о потребленном трафике для конкретного пользователя;
- подсчет общей стоимости реализованного трафика (использовать интерфейсы пространства имен GenericMath);
- поиск клиента, заплатившего наибольшую стоимость за услуги.
- подсчитывать общие выплаты по конкретному тарифу

### **10. Предметная область: Интернет магазин.**

В информационной системе хранятся данные о товарах. При оформлении заказа регистрируется количество товара. Клиент может заказать разные товары.

Система должна позволять выполнять следующие задачи:

- ввод информации о товарах;
- регистрация заказа клиента на покупку определенного товара;
- после ввода фамилии покупателя предоставление списка заказанных им товаров;
- после ввода фамилии покупателя вычислять суммы заказа (использовать интерфейсы пространства имен GenericMath).
- вычислять, на какую сумму было продано конкретного товара

## 2. Лабораторная работа №2

### События, исключения

**Цель работы:** знакомство с механизмом событий и исключений в языке C#.

**Задача работы:** научиться генерировать исключения, а также корректно обрабатывать исключительные ситуации; использовать события для уведомления других классов о совершенном действии.

**Время выполнения работы:** 4 часа (2 занятия)

**Результат выполнения работы:** программа, обеспечивающая функционал согласно заданию.

#### 2.1. Задание для всех вариантов

**Используйте проект из лабораторной работы №1**

а) В классе **MyCustomCollection<T>** реализовать возможность обхода коллекции оператором **foreach** (реализовать интерфейс **IEnumerable<T>**).

б) В классе **MyCustomCollection<T>** генерировать исключения:

**IndexOutOfRangeException** – в индексаторе коллекции при обращении к несуществующему индексу

**свое собственное исключение** – в методе *void Remove(T item)*, если объект *item* отсутствует в коллекции

с) В сущностном классе описать события согласно индивидуальному заданию. Для событий использовать стандартные делегаты. Событие должно передавать с помощью аргумента информацию о совершенном действии.

д) В папке **Entities** опишите класс **Journal**, который будет хранить информацию о событиях (см. п.3) - описание события, имя сущности согласно индивидуальному заданию. Предусмотреть публичные методы для регистрации событий. Класс **Journal** также должен позволять при необходимости выводить в консоль список всех зарегистрированных событий.

е) В классе **Program** создать объект класса **Journal** и подписать его на событие согласно индивидуальному заданию.

ф) В классе **Program** подписаться на событие согласно индивидуальному заданию. Обработчик события – **лямбда-выражение** – вывод на консоль информации о событии.

г) Продемонстрировать обработку исключений, возникающих при работе с коллекцией **MyCustomCollection<T>**.

## 2.2. Индивидуальные задания

### 1. *Предметная область: АТС.*

АТС генерирует события:

- при изменении списка тарифов или списка абонентов (на это событие должен быть подписан класс **Journal**)
- при совершении звонка абонентом (на это событие должен быть подписан класс **Program**)

### 2. *Предметная область: Вокзал.*

Вокзал генерирует события:

- при изменении списка тарифов или списка пассажиров (на это событие должен быть подписан класс **Journal**)
- при покупке билета пассажиром (на это событие должен быть подписан класс **Program**)

### 3. *Предметная область: ЖЭС.*

ЖЭС генерирует события:

- при изменении списка тарифов или списка жильцов (на это событие должен быть подписан класс **Journal**)
- при покупке услуги жильцом (на это событие должен быть подписан класс **Program**)

### 4. *Предметная область: Аэропорт.*

Аэропорт генерирует события:



- при изменении списка тарифов или списка пассажиров (на это событие должен быть подписан класс Journal)
- при покупке билета пассажиром (на это событие должен быть подписан класс Program)

**5. Предметная область: Банк.**

Вокзал генерирует события:

- при изменении списка вкладов или списка клиентов (на это событие должен быть подписан класс Journal)
- при регистрации вклада клиентом (на это событие должен быть подписан класс Program)

**6. Предметная область: Отдел расчета зарплаты.**

Отдел генерирует события:

- при изменении списка тарифов или списка работников (на это событие должен быть подписан класс Journal)
- при выполнении работы работником (на это событие должен быть подписан класс Program)

**7. Предметная область: Фирма грузоперевозок.**

Фирма генерирует события:

- при изменении списка тарифов или списка клиентов (на это событие должен быть подписан класс Journal)
- при заказе перевозки клиентом (на это событие должен быть подписан класс Program)

**8. Предметная область: Гостиница.**

Гостиница генерирует события:

- при изменении списка номеров или списка клиентов (на это событие должен быть подписан класс Journal)
- при заказе номера клиентом (на это событие должен быть подписан класс Program)

**9. Предметная область: Интернет оператор.**

Провайдер генерирует события:

- при изменении списка тарифов или списка клиентов (на это событие должен быть подписан класс Journal)
- при использовании трафика клиентом (на это событие должен быть подписан класс Program)

**10. Предметная область: Интернет магазин.**

Магазин генерирует события:

- при изменении списка товаров или списка клиентов (на это событие должен быть подписан класс Journal)
- при заказе товара клиентом (на это событие должен быть подписан класс Program)

## 3. Лабораторная работа №3

### Стандартные коллекции. технология LINQ

**Цель работы:** знакомство с коллекциями из пространства имен System.Collections.Generic. Знакомство с технологией LINQ-to-Object.

**Задача работы:** научиться использовать стандартные коллекции в соответствии с поставленной задачей. Написать запросы LINQ для поиска, фильтрации данных.

**Время выполнения работы:** 4 часа (2 занятия)

**Результат выполнения работы:** программа, обеспечивающая функционал согласно заданию.

#### 3.1.Задание

а) Создать новый проект, в котором реализовать задание из лабораторных работ №1 и №2 с использованием **стандартных обобщенных коллекций** согласно индивидуальному заданию.

б) Выполнить индивидуальные задания **с помощью запросов LINQ**. Результаты запросов вывести в консоль.

Стиль написания запросов (выражения запросов или расширяющие методы) – по желанию.

#### 3.2. Индивидуальные задания

##### 1. *Предметная область: АТС.*

Для списка тарифов АТС использовать Dictionary<string, T>.

Для списка клиентов использовать List<T>

Для списка тарифов клиента использовать List<T>.

События (см. задание лабораторной работы №2) описать и вызывать в классе АТС.

Описать методы :

Получение списка **названий** всех тарифов, отсортированного по стоимости

Получение общей стоимости всех выполненных на АТС звонков.

Получение общей стоимости всех сделанных клиентом звонков в соответствии с действующими тарифами;

Получение **имени** клиента, заплатившего максимальную сумму. Если таких клиентов несколько, получить имя первого в списке

Получение количества клиентов, заплативших больше определенной суммы (использовать Aggregate)

Получение клиентом списка сумм, заплаченных по каждому тарифу (использовать GroupBy).

## **2. Предметная область: Вокзал.**

Для списка тарифов вокзала использовать Dictionary<string, T>.

Для списка пассажиров использовать List<T>

Для списка билетов пассажира использовать List<T>.

События (см. задание лабораторной работы №2) описать и вызывать в классе Вокзал.

Описать методы :

Получение списка **названий** всех тарифов, отсортированного по стоимости

Получение общей стоимости всех купленных на вокзале билетов.

Получение общей стоимости всех купленных пассажиром билетов в соответствии с действующими тарифами;

Получение **имени** пассажира, заплатившего максимальную сумму. Если таких пассажиров несколько, получить имя первого в списке

Получение количества пассажиров, заплативших больше определенной суммы (использовать Aggregate)

Получение пассажиром списка сумм, заплаченных по каждому направлению (использовать GroupBy).

## **3. Предметная область: ЖЭС.**

Для списка услуг ЖЭС использовать Dictionary<string, T>.

Для списка жильцов использовать List<T>

Для списка услуг клиента использовать List<T>.

События (см. задание лабораторной работы №2) описать и вызывать в классе ЖЭС.

Описать методы :

Получение списка **названий** всех услуг, отсортированного по стоимости

Получение общей стоимости всех выполненных услуг ЖЭС.

Получение общей стоимости всех услуг, заказанных жильцом в соответствии с действующими тарифами;

Получение **имени** жильца, заплатившего максимальную сумму. Если таких жильцов несколько, получить имя первого в списке

Получение количества жильцов, заплативших больше определенной суммы (использовать Aggregate)

Получение жильцом списка сумм, заплаченных по каждой услуге (использовать GroupBy).

#### **4. Предметная область: Аэропорт.**

Для списка тарифов аэропорта использовать Dictionary<string, T>.

Для списка пассажиров использовать List<T>

Для списка билетов пассажира использовать List<T>.

События (см. задание лабораторной работы №2) описать и вызывать в классе Аэропорт.

Описать методы :

Получение списка **названий** всех тарифов, отсортированного по стоимости

Получение общей стоимости всех купленных в аэропорту билетов.

Получение общей стоимости всех купленных пассажиром билетов в соответствии с действующими тарифами;

Получение **имени** пассажира, заплатившего максимальную сумму. Если таких пассажиров несколько, получить имя первого в списке

Получение количества пассажиров, заплативших больше определенной суммы (использовать Aggregate)

Получение пассажиром списка сумм, заплаченных по каждому направлению (использовать GroupBy).

#### **5. Предметная область: Банк.**

Для списка тарифов банка использовать Dictionary<string, T>.

Для списка клиентов использовать List<T>

Для списка вкладов клиента использовать List<T>

События (см. задание лабораторной работы №2) описать и вызывать в классе Банк

Описать методы :

Получение списка **названий** всех тарифов, отсортированного по величине процента

Получение общей величины процентов всех оформленных в банке вкладов.

Получение общей величины всех вкладов, сделанных в банке;

Получение **имени** клиента, заплатившего максимальную сумму процентов. Если таких клиентов несколько, получить имя первого в списке

Получение количества клиентов, заплативших больше определенной суммы (использовать Aggregate)

Получение списка сумм, заплаченных по каждому клиенту (использовать GroupBy).

**6. Предметная область: Отдел расчета зарплаты.**

Для списка работ отдела использовать Dictionary<string, T>.

Для списка работников использовать List<T>

Для списка работ работника использовать List<T>.

События (см. задание лабораторной работы №2) описать и вызывать в классе «Отдел расчета зарплаты»

Описать методы :

Получение списка **названий** всех тарифов работ, отсортированного по стоимости

Получение общей стоимости всех выполненных работ.

Получение общей стоимости всех выполненных работ работника в соответствии с действующими тарифами;

Получение **имени** работника, заработавшего максимальную сумму. Если таких клиентов несколько, получить имя первого в списке

Получение количества работников, получивших больше определенной суммы (использовать Aggregate)

Получение работником списка сумм, полученных по каждому виду работ (использовать GroupBy).

**7. Предметная область: Фирма грузоперевозок.**

Для списка тарифов фирмы использовать Dictionary<string, T>.

Для списка клиентов использовать List<T>

Для списка заказов клиента использовать List<T>.

События (см. задание лабораторной работы №2) описать и вызывать в классе «Фирма».

Описать методы :

Получение списка **названий** всех тарифов, отсортированного по стоимости

Получение общей стоимости всех выполненных фирмой перевозок.

Получение общей стоимости всех перевозок, заказанных клиентом в соответствии с действующими тарифами;

Получение **имени** клиента, заплатившего максимальную сумму. Если таких жильцов несколько, получить имя первого в списке

Получение количества клиентов, заплативших больше определенной суммы (использовать Aggregate)

Получение клиентом списка сумм, заплаченных по каждому тарифу (использовать GroupBy).

**8. Предметная область: Гостиница.**

Для списка номеров гостиницы использовать Dictionary<int, T>.

Для списка клиентов использовать List<T>

Для списка номеров клиента использовать List<T>

События (см. задание лабораторной работы №2) описать и вызывать в классе Гостиница.

Описать методы :

Получение списка **номеров комнат**, отсортированного по стоимости

Получение общей стоимости всех забронированных в гостинице номеров.

Получение **имени** клиента, заплатившего максимальную сумму. Если таких пассажиров несколько, получить имя первого в списке

Получение количества клиентов, заплативших больше определенной суммы (использовать Aggregate)

Получение списка, показывающего, сколько номеров имеется в гостинице по каждой ценовой категории (использовать GroupBy).

**9. Предметная область: Интернет оператор.**

Для списка тарифов оператора использовать Dictionary<string, T>.

Для списка клиентов использовать List<T>

Для списка тарифов клиента использовать List<T>.

События (см. задание лабораторной работы №2) описать и вызывать в классе «Интернет оператор»

Описать методы :

Получение списка **названий** всех тарифов, отсортированного по стоимости

Получение общей стоимости всего трафика оператора.

Получение **имени** клиента, заплатившего максимальную сумму. Если таких клиентов несколько, получить имя первого в списке

Получение количества клиентов, заплативших больше определенной суммы (использовать Aggregate)

Получение списка, показывающего, сколько клиентов зарегистрировано по каждому тарифу (использовать GroupBy).

**10. Предметная область: Интернет магазин.**

Для списка товаров магазина использовать Dictionary<string, T>.

Для списка клиентов использовать List<T>

Для списка товаров клиента использовать List<T>.

События (см. задание лабораторной работы №2) описать и вызывать в классе «Интернет-магазин».

Описать методы :

Получение списка **названий** всех товаров, отсортированного по стоимости

Получение общей стоимости всех купленных товаров магазина.

Получение общей стоимости всех товаров, заказанных клиентом в соответствии с действующими тарифами;

Получение **имени** клиента, заплатившего максимальную сумму. Если таких жильцов несколько, получить имя первого в списке

Получение количества клиентов, заплативших больше определенной суммы (использовать Aggregate)

Получение клиентом списка сумм, заплаченных по каждому товару (использовать GroupBy).



## 4. Лабораторная работа №4

Файлы и потоки данных

**Цель работы:** знакомство с классами пространства имен System.IO.

**Задача работы:** Научиться выполнять основные операции над объектами файловой системы. Научиться выполнять чтение/запись из/в файл.

**Время выполнения работы:** 4 часа (2 занятия)

**Результат выполнения работы:** программа, обеспечивающая функционал согласно заданию.

### 4.1. Задание

- a) Создать новый проект
- b) Описать класс согласно индивидуальному заданию, содержащий любые свойства типа *int* и *bool* и свойство *Name* (имя) типа *string*

- c) Описать интерфейс *FileService*:

```
interface IFileService
{
    IEnumerable<T> ReadFile(string fileName);
    void SaveData(IEnumerable<T> data, string fileName);
}
```

- d) Описать класс *FileService*, реализующий интерфейс *IFileService* для класса вашей предметной области.

- Метод *ReadFile* – **именованный итератор** (оператор *yield return*), считывающий данные из **бинарного** файла с именем *fileName*

- Метод *SaveData* сохраняет коллекцию *data* в **бинарный** файл с именем *filename*. Если файл с таким именем уже существует, предварительно удалить существующий файл.

- Запись и чтение файла выполнить с помощью *BinaryReader/BinaryWriter*

- Файловый поток после чтения/записи должен быть уничтожен (использовать оператор *using*)
  - Обработать исключения, генерируемые объектами BinaryReader/BinaryWriter (только эти исключения) при выполнении операций чтения/записи.
- е) Описать класс `MyCustomComparer<T>:IComparer<T>`, где T - класс вашей предметной области, позволяющий сравнивать объекты класса по свойству Name
- ф) В классе Program:
- Создать папку с именем XXX\_Lab4, где XXX – ваша фамилия
  - Заполните папку 10-ю пустыми файлами с расширениями .txt, .rtf, .dat, .inf (выбирать случайным образом в коде). Имена файлов генерировать случайно с помощью класса Path.
  - Прочитать содержимое папки и вывести список файлов в виде:  
«Файл: <имя файла> имеет расширение <расширение файла>»
  - Создать коллекцию объектов класса вашей предметной области.
- Заполнить коллекцию 5-6 объектами
- С помощью класса FileService записать в файл коллекцию объектов класса Employee.
  - Переименовать файл.
  - Создать пустую коллекцию объектов класса вашей предметной области. С помощью класса FileService заполнить коллекцию данными из файла с новым именем.
  - Отсортировать полученную коллекцию с помощью запроса LINQ (использовать класс `MyCustomComparer`)
  - Вывести в консоль содержимое исходной коллекции и отсортированной коллекции, прочитанной из файла

- Отсортировать коллекцию с помощью запроса LINQ по любому свойству, отличному от Name. Для метода сравнения использовать **лямбда-выражение**.
- Вывести в консоль содержимое отсортированной коллекции

### 1.3.Индивидуальные задания

- 1) *Предметная область – жильцы дома.*
- 2) *Предметная область – компьютеры.*
- 3) *Предметная область – пассажиры.*
- 4) *Предметная область – сотрудники предприятия.*
- 5) *Предметная область – автомобили.*
- 6) *Предметная область – клиенты компании*
- 7) *Предметная область – товары магазина*
- 8) *Предметная область – предметы искусства*
- 9) *Предметная область –багаж пассажиров*
- 10) *Предметная область – продукты питания.*
- 11) *Предметная область – музыкальные произведения.*
- 12) *Предметная область – студенты.*
- 13) *Предметная область – млекопитающие.*
- 14) *Предметная область – слушатели курсов.*
- 15) *Предметная область – игрушки.*
- 16) *Предметная область – вклады банка.*
- 17) *Предметная область – билеты в кино*
- 18) *Предметная область – пациенты больницы.*
- 19) *Предметная область – участники соревнований.*
- 20) *Предметная область – супергерои.*