## Оценка характеристик программы

### Размер класса

### Кол-во операций переопределяемых классом

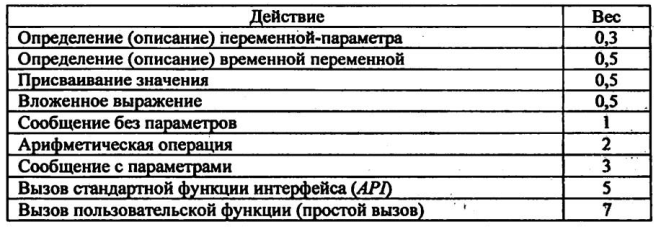
### Кол-во операций добавленных классом

### Индекс специализации

### Средний размер операций

* 2 сообщения классу *Person*
* 3 сообщения классу *IEnumerable*
* 3 сообщения классу *Console*
* 1 сообщение классу *String*
* 1 сообщение классу *List*
* 1 сообщение классу *IEnumerable*

### Сложность операций



public Person(ulong passportNumber, string lastName)

{

PassportNumber = passportNumber;

LastName = lastName;

}

public override string ToString()

{

return $"{PassportNumber} {LastName}";

}

public Resident(ulong passportNumber, string lastName, string hotel, uint daysStay)

: base(passportNumber, lastName)

{

Hotel = hotel;

DaysStay = daysStay;

}

public override string ToString()

{

return $"{base.ToString()} {Hotel} {DaysStay}";

}

public static int DaysStayOfPersonInHotel(this IEnumerable<Resident> residents, string person\_last\_name, string hotel\_name)

{

return (int)residents

.Where(p => p.LastName == person\_last\_name)

.Where(p => p.Hotel == hotel\_name)

.Sum(p => p.DaysStay);

}

static void Main()

{

List<Resident> table = new()

{

new (1\_111\_111, "Иванов", "Айсберг", 5),

new (2\_222\_222, "Петров", "Элегант", 7),

new (2\_222\_222, "Петров", "Элегант", 3),

new (3\_333\_333, "Сидоров", "Айсберг", 1),

new (2\_222\_222, "Петров", "Айсберг", 3),

};

Console.WriteLine(string.Join(Environment.NewLine, table));

Console.Write("Введите название отеля : ");

string hotel = Console.ReadLine();

int days = table.DaysStayOfPersonInHotel(person, hotel);

Console.WriteLine(days);

}

### Кол-во параметров на операцию

### Кол-во описаний сценариев

### Кол-во ключевых классов

### Кол-во подсистем

* определяющая проживающего человека как сущность
* определяющая запросы к списку проживающих
* основная упраляющая