Отчёт по практикуму

«объектно-ориентированное

программирование»

Студент: Бутиков Илья Иванович

Группа: 341

Москва 2021

Оглавление

1. Постановка задачи

Рассматривается работа аэропорта с N взлетно-посадочными полосами (2≤ N ≤ 10). Необходимо создать систему-диспетчер, обрабатывающую заявки на взлет и посадку самолетов нескольких авиакомпаний, и провести эксперименты с ней, программно смоделировав поток заявок.

Заявки на взлет и посадку формируются на основе действующего суточного расписания полетов в данном аэропорту (оно включает расписание вылетов и расписание прилетов самолетов) с учетом возможных отклонений от расписания (из-за задержек загрузки топлива и по другим причинам). Отклонение от расписания моделируется как случайная величина, имеющая нормальное распределение в некотором интервале, но не более, чем от -120 до 120 минут. Для вылетов возможны только задержки, для посадок – как задержки, так и досрочные прилеты. Фактическое время начала взлета или посадки самолета определяется как время по расписанию, измененное на случайную величину отклонения, а также на время ожидания свободной полосы для взлета/посадки.

Модель должна содержать 3 типа самолетов: грузовой, пассажирский и бизнес-джет. Длительность взлета/посадки зависит от типа самолета. Между последовательными взлетами/посадками самолетов на одной полосе предусмотрены небольшие временные промежутки (от 0 до 10 минут).

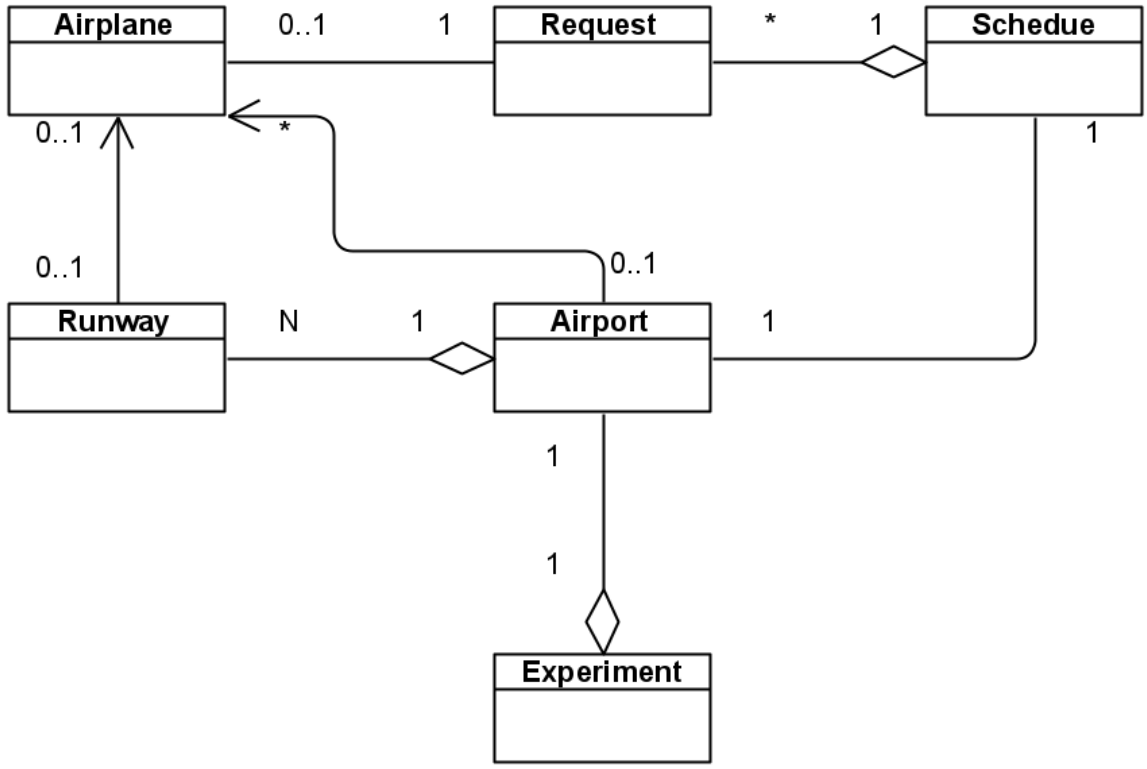
Цель моделирования функций диспетчера взлетно-посадочных полос – определение оптимального количества полос для безопасного обслуживания взлетов/посадок. Одним из безопасных режимов считается разделение всех полос на непересекающиеся множества – полосы для взлета и полосы для посадки. Другой безопасный режим – приоритет самолетов, идущих на посадку, над самолётами, идущими на взлёт. Период моделирования – сутки.

В параметры моделирования следует включить: число N взлетно-посадочных полос, исходное расписание полетов в данном аэропорту, диапазон разброса случайной величины отклонения от расписания, шаг моделирования (интервал времени от 1 до 30 минут), режим разделения полос и количество полос для взлета и посадки в данном режиме, а также время суток, с которого начинается моделирование работы диспетчера.

Визуализация моделируемого процесса должна предусматривать показ состояния (занята/свободна) каждой взлетно-посадочной полосы и очереди самолетов на взлет и посадку, а также график уже произведенных взлетов и посадок с указанием времени и взлетной полосы. В ходе и по окончании моделирования следует вывести статистическую информацию: общее

количество обслуженных заявок, общее количество обслуженных заявок на взлёт, общее количество обслуженных заявок на посадку, динамику изменения задержки на взлёт и посадку, длину очереди на взлёт и посадку, среднюю занятость взлетно-посадочных полос.

2. Диаграмма классов



3. Текстовые спецификации интерфейса

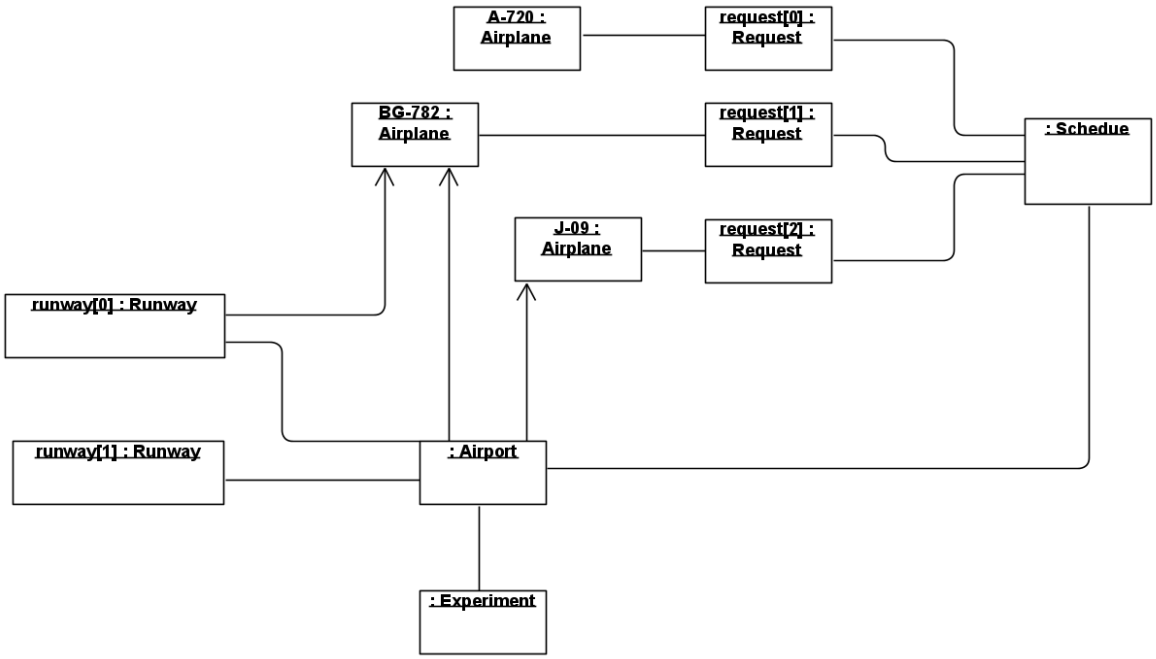
```
    enum Direction { Landing, Takeoff}//посадка, взлёт
    class Request
    {
        public string AirplaneName { get; private set; }
        public string CompanyName { get; private set; }

        public AirType airType { get; private set; }
        public Direction dir { get; private set; }

        public int TimeSchedue { get; private set; }//время по расписанию
        public int TimeReal { get; private set; } = -1; //время начала
обработки заявки
        public int TimeEvent { get; private set; }// время по расписанию

        public Airplane airplane { get; private set; } = null;
```

```
        public Request(int SchedueTime, string AirplaneName, string
CompanyName,
                       Direction dir, AirType airType, int EventTime);
        public void ServiceStarted(int Time); //когда самолёту назначается
полоса
        public void Tick(int WorldTime, Airport airport);
    }

    public enum AirType { Cargo, Passenger, Jet }
    public enum State { Waiting, RunwayIn, TakingOff,
                        AirWaiting, Landing, RunwayOut, Done}

    class Airplane
    {
        public static int TimeMoveOnRunway = 5; //время захода на полосу и
ухода с полосы

        public string Name {  get; private set; }
        public string CompanyName { get; private set; }

        public AirType Type { get; private set; }
        public State state { get; private set; }
        public int MoveTime { get; private set; }//время, необходимое для
взлета/посадки
        public int Timer { get; private set; }
        public int Runway { get; private set; } = -1;
        public Request SummonerRequest { get; private set; }


        public Airplane(string name, string CompanyName, AirType type,
                        Direction dir,  Request request);
        public void SetRunway(int runway);//назначение на полосу
        public void Tick();
    }

    class Schedue
    {
        Random rnd;//для генерации числа из нормального распределения и для
значения мирового времени
        int StartRequest = 0;//для оптимизации, чтобы не дёргать обработанные
запросы

        public List<Request> requests { get; private set; }
        public Schedue(string fileName, Random rnd, int StartTime, int
DelayMin, int DelayMax);

        public void Tick(int WorldTime, Airport airport);
      //генерация значения случайной величины из нормального распределения
        private double GenerateNormalDistribution(double a, double sigm);

      //чтение расписания из текстового файла
void TextFileCreateSchedue(string fileName, int StartTime, int DelayMin, int
DelayMax);
      //чтение расписания из файла exel
```

```
        void ExelCreateSchedue(string fileName,int StartTime, int DelayMin,
int DelayMax);
    }

    class Runway
    {
        public bool forTakeoff { get; private set; } = true;
        public bool forLanding { get; private set; } = true;


        int TimeInterval, CurrentTimeInterval = 0;//минимальное время между
взлетами/посадками
        public Airplane tmpAirplane { get; private set; } = null;
        public delegate void newDoneRequest(int numberRunway, Direction dir);
        public event newDoneRequest SuccessRequest;
        public Runway(bool forTakeoff, bool forLanding, int TimeInterval);

        public void SetAirplane(Airplane airplane, int i);//назначение
самолёта на полосу
        public void Clear();//самолёт освобождает полосу
        public bool Ready();//готова ли полоса к приему новых самолётов?
        public void Tick();
        }
    }


    class Airport
    {
        bool ModSepRunway;

        public Runway[] runway { get; private set; }
        int CountRunway;
        int CountLandingRunway;
        public Schedue schedue { get; private set; }

        public Queue<Airplane> TakeoffQueue { get; private set; }
        public Queue<Airplane> LandingQueue { get; private set; }

        public Airport( bool ModSepRunway, int CountRunway, int
CountLandingRunway,
                        int TimeInterval, int DelayMin, int DelayMax,
                        string fileName, Random rnd, int StartTime);

        public void NewAirplane(Airplane pl);//добавление самолёта в очередь
        void DistributionRunways(); // распределить полосы
        public void Tick(int WorldTime);
    }

    class Experiment
    {

        int StopTime = 24 * 60;//конец моделирования
        public int TimeStep { get; set; } = 5;//шаг моделирования
        public int CurrentTime { get; private set; } = -1;//чтобы обработать
первую заявку
```

```
        public Airport airport { get; private set; }
        Random rnd;
        public Experiment(int TimeStep,int StartTime,
            bool ModSepRunway, int CountRunway,  int CountLandingRunway,
            int DelayMin,int DelayMax, int TimeInterval,
            string fileName);

        public bool Tick();
        public bool NextStep();// вызовы Tick() в количестве, равному шагу
моделирования
        public bool ToEnd();// вызовы Tick() до конца моделирования
    }
```

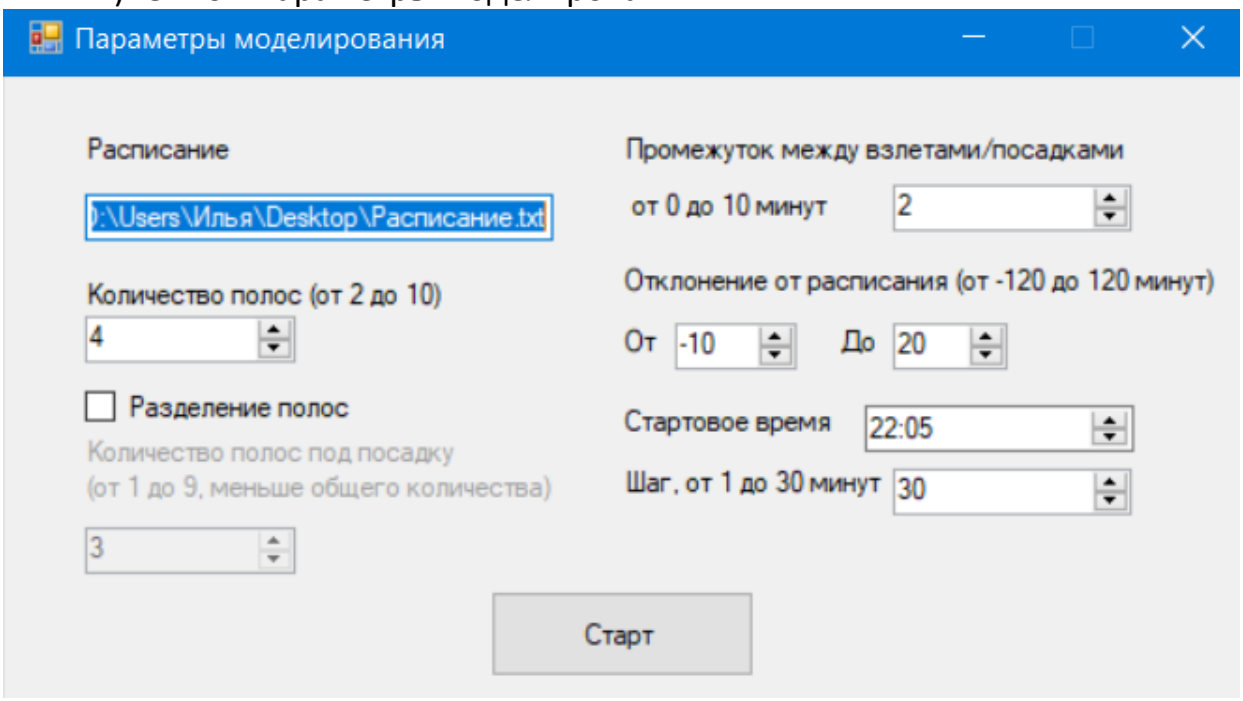
4. Диаграмма объектов



5. Инструментальные средства
    1) Язык разработки – C#
    2) Среда разработки – Microsoft Visual Studio .NET Framework 4.7.2
    3) Используемые библиотеки
        - System;
        - System.Collections.Generic;
        - Microsoft.Office.Interop.Excel;
        - System.IO;
        - System.Linq;
        - System.Windows.Forms;
        - System.Windows.Forms.DataVisualization.Charting;
        - System.Drawing;
        - System.Linq;

6. Описание файловой структуры системы
    1) Program.cs – главный файл, запуск программы
    2) MainDisplayForm.cs – интерфейс основного окна
    3) Form1.cs – интерфейс окна параметров
    4) Experiment.cs – описание класса Experiment
    5) Airport.cs – описание класса Airport
    6) Runway.cs – описание класса Runway
    7) Schedue.cs – описание класса Schedue
    8) Request.cs – описание класса Request
    9) Airplane.cs – описание класса Airplane
    10) Resources.resx – файл с изображенями самолётов

7. Пользовательский интерфейс
    1) Окно «Параметры моделирования»



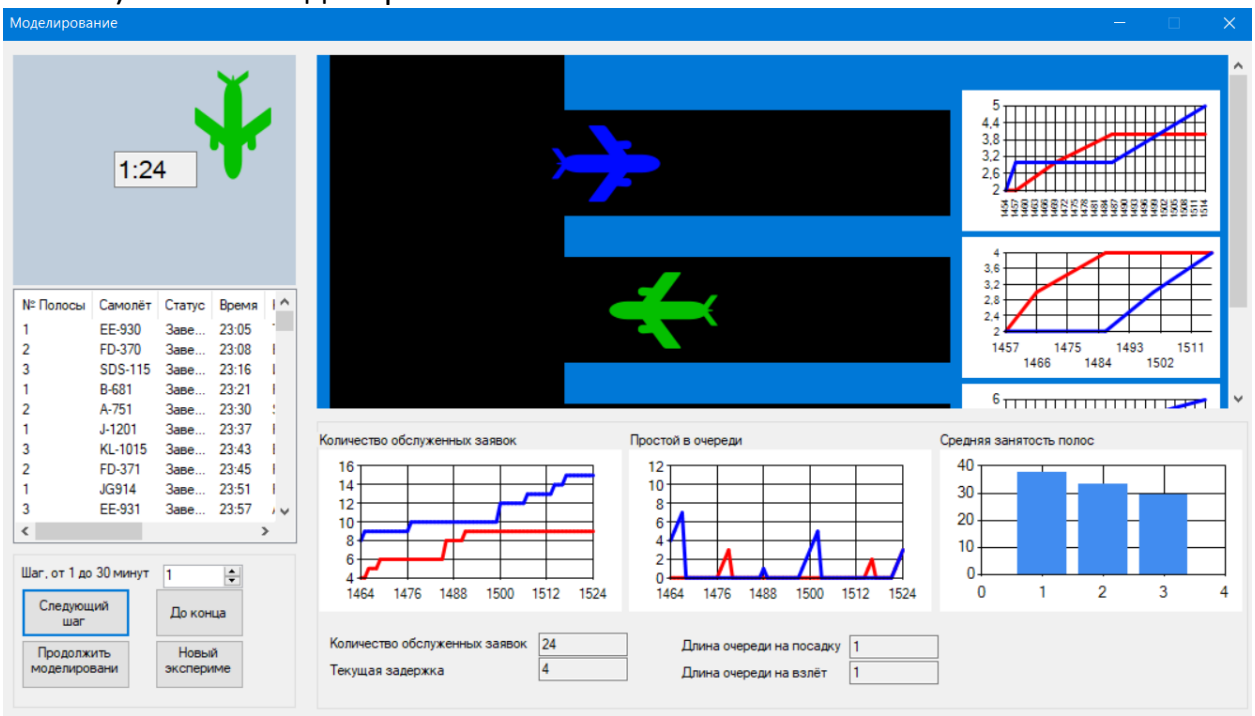
Параметры моделирования:

- Суточное расписание – в виде файла txt или excel.
- Время начала моделирования – стартовое время модели. Если время заявки, указанное в файле, меньше времени начала моделирования, то к нему добавляется 24 часа.
- Количество полос – от 2 до 10
- Режим разделения полос – если активно, часть полос отдается только под посадку, остальные – только для взлета. Иначе – с каждой полосы самолет может как взлетать, так и садиться на неё.

- Количество полос под посадку – от 1 до 9, меньше общего количества полос. Доступно только при выбранном режиме разделения полос. Определяет количество полос, которые будут отданы для посадки. Оставшиеся полосы будут отданы под взлёт.
- Промежуток между взлётами/ посадками – от 0 до 10 минут – минимальное время между окончанием взлета/посадки одного самолёта на полосу и началом взлета/посадки с этой же полосы следующего самолёта. Моделирует отрезок время тех. обслуживания полосы после взлёта/посадки самолета.
- Отклонения от расписания – от -120 до 120 минут – границы временного интервала, в которых находится отклонение начала взлёта/посадки самолёта от расписания. Для взлетающих самолётов минимальная граница рассчитывается как максимум из указанного минимума и нуля.
- Шаг моделирования – от 1 до 30 минут – интервал времени модели между отрисовками кадров.

Кнопка «Старт» - запускает моделирование с выбранными параметрами.

2) Окно «Моделирование»



Кнопки:

- Следующий шаг – вычислить параметры модели через промежуток времени модели, равный размеру шага, и перерисовать экран.

- До конца – вычислить параметры модели на конец моделирования, и перерисовать экран.
- Прервать моделирование/Продолжить моделирование – прекратить/возобновить автоматическое обновление модели согласно размеру шага моделирования.
- Новый эксперимент – прерывание текущего эксперимента, открытие окна выбора параметров моделирования.

Поле «Шаг» - установить текущий размер шага моделирования.

В левом верхнем углу можно увидеть часы, отображающие текущее время модели.
В сером прямоугольнике отображается очередь самолётов на посадку.
Под ним – Расписание, с указанием номера полосы, на которую заходил или заходит самолёт, и временем задержки.

В центре нарисованы полосы, общая полоса, соединяющая остальные, графики к полосам, самолеты, двигающиеся по полосам. Цвет самолёта указывает на его тип: зелёный – грузовой, синий – пассажирский, желтый – бизнес-джет.

Графики:

- Количество обслуженных заявок – показывает динамику обслуженных заявок за последний час модели.
- Простой в очереди – показывает динамику изменения максимальной задержки для взлёта и посадки за последний час модели.
- Средняя занятость полос – показывает отношение количества заявок, обслуженных на полосе к общему количеству обслуженных заявок.
- Графики около полос – для каждой полосы показывает динамику обслуженных заявок.

Красной линией обозначены заявки на взлёт, синей – на посадку.

Поля:

- Количество обслуженных заявок – количество обслуженных заявок на текущее время модели.
- Текущая задержка – максимум из текущей задержки на взлет и текущей задержки на посадку.
- Длина очереди на посадку – количество самолётов, ожидающих распределения на полосу для посадки.
- Длина очереди на взлёт – количество самолётов, ожидающих распределения на полосу для взлёта.

8. Описание проведённых экспериментов

Было составлено расписание из 288 заявок, 154 на посадку и 134 на взлёт, с 5 минутным интервалом между каждой заявкой.
В ходе экспериментов было выявлено:

- Увеличение промежутка между взлётами и посадками увеличивает задержку
    - Для нулевого отклонения, чтобы задержка была нулевой, хватает 3 полос при промежутке между посадками == 0
    - И 7 при промежутке между посадками == 10
    - 6 полос при промежутке между посадками == 10 дают задержку около получаса.
- Режим разделения полос в среднем менее эффективен (общая задержка больше), чем основной режим.
    - Зато он позволяет уменьшить задержку на взлёт, не давая большому потоку самолётов на посадку занимать все полосы.
- 2 полосы недостаточны для обслуживания данного расписания, не зависимо от остальных параметров
    - При режиме разделения полос задержка на посадку может достичь 2 часов, что привело бы к падению большинства самолётов.
    - При основном режиме задержка на взлёт может вырасти до 9-10 часов, что ведёт к большим убыткам компаний, и, следственно, аэропорта.
- Увеличение диапазон разброса случайной величины отклонения от расписания приводит к увеличению «скачкообразности» графика простоя