|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | **6** |

**Название:**

«Моделирование системы с очередями»

**Дисциплина:** Моделирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-76Б |  |  | А. А. Петрова |
|  | (Группа) |  |  | (И.О. Фамилия) |
|  |  |  |  |  | |  |
| Преподаватель |  |  |  | И. В. Рудаков |
|  |  |  |  | (И.О. Фамилия) |

2022 г.

**Задание**

Для сдачи лабораторных работ по предмету приходят студенты через интервалы времени 2 ± 1 минуты и встают в две очереди к двум соответствующим преподавателям. Первый преподаватель принимает одного студента за 4 ± 2 минуты, а второй – за 20 ± 5 минут. Студенты встают в очередь к первому преподавателю с заданной вероятностью. После сдачи студенты из обеих очередей встают в одну очередь на защиту. Защита одного студента принимается за 5 минут. Смоделировать процесс сдачи лабораторных работ при 100 студентах.

**Математическая формализация**

Эндогенные переменные: время принятия лабораторной i-ым преподавателем и время принятия защиты.

Экзогенные переменные: число принятых студентов у первого преподавателя и второго.

Ниже представлена схема модели в терминах СМО.

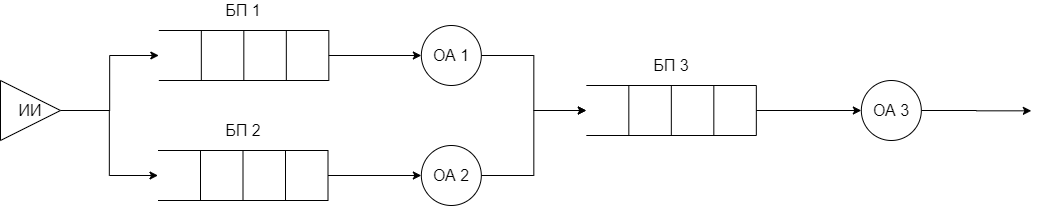


Рисунок 1: модель в терминах СМО

Где ИИ – это источник информации (студент с лабораторной), БП – буферная память (очередь), а ОА – обслуживающий аппарат (преподаватели и защита).

**Реализация**

В листингах ниже представлена реализация генератора заявок, работы преподавателей и защиты.

Листинг 1: генератор заявок

|  |
| --- |
| **class** **TimeProcessor:**  **def** \_\_init\_\_**(**self**,** time\_distribution**):**  self**.**time\_distribution **=** time\_distribution  self**.**remaining\_time **=** 0  **def** update\_time**(**self**):**  **if** self**.**remaining\_time **>** 0**:**  self**.**remaining\_time **-=** TIME\_DELTA  **if** self**.**remaining\_time **<=** 1e-5**:**  self**.**remaining\_time **=** self**.**time\_distribution**.**get\_time**()**  **return** Request**()**  **return** **None** |

Листинг 2: преподаватель

|  |
| --- |
| **class** **Teacher:**  **def** \_\_init\_\_**(**self**,** storage**,** recipient**,** time\_distribution**):**  self**.**time\_distribution **=** time\_distribution  self**.**requests\_storage **=** storage  self**.**recipient **=** recipient  self**.**remaining\_time **=** 0  self**.**is\_busy **=** **False**  self**.**processing\_request **=** **None**  self**.**max\_queue\_len **=** 0  self**.**work\_time **=** 0  **def** update\_time**(**self**):**  self**.**remaining\_time **-=** TIME\_DELTA  **if** **len(**self**.**requests\_storage**)** **>** self**.**max\_queue\_len**:**  self**.**max\_queue\_len **=** **len(**self**.**requests\_storage**)**  **if** self**.**is\_busy **and** self**.**remaining\_time **<=** 1e-5**:**  self**.**recipient**.**append**(**self**.**processing\_request**)**  self**.**is\_busy **=** **False**  self**.**processing\_request **=** **None**  **if** **not** self**.**is\_busy **and** **len(**self**.**requests\_storage**)** **!=** 0**:**  self**.**processing\_request **=** self**.**requests\_storage**.**pop**(**0**)**  self**.**remaining\_time **=** self**.**time\_distribution**.**get\_time**()**  self**.**work\_time **+=** self**.**remaining\_time  self**.**is\_busy **=** **True** |

Листинг 3: защита

|  |
| --- |
| **class** **Defense:**  **def** \_\_init\_\_**(**self**,** requests\_storage**,** time\_distribution**):**  self**.**requests\_storage **=** requests\_storage  self**.**time\_distribution **=** time\_distribution  self**.**is\_busy **=** **False**  self**.**processing\_request **=** **None**  self**.**remaining\_time **=** 0  self**.**max\_queue\_len **=** 0  **def** update\_time**(**self**):**  **if** self**.**remaining\_time **!=** 0**:**  self**.**remaining\_time **-=** TIME\_DELTA  **if** **len(**self**.**requests\_storage**)** **>** self**.**max\_queue\_len**:**  self**.**max\_queue\_len **=** **len(**self**.**requests\_storage**)**  **if** self**.**is\_busy **and** self**.**remaining\_time **<=** 1e-5**:**  self**.**is\_busy **=** **False**  self**.**processing\_request **=** **None**  **return** FINISH\_PROCESS\_REQUEST  **if** **not** self**.**is\_busy **and** **len(**self**.**requests\_storage**)** **!=** 0**:**  self**.**processing\_request **=** self**.**requests\_storage**.**pop**(**0**)**  self**.**remaining\_time **=** self**.**time\_distribution**.**get\_time**()**  self**.**is\_busy **=** **True** |

**Результаты работы**

Ниже представлены результаты работы программы для 100 студентов с разными вероятностями выбора первого преподавателя.

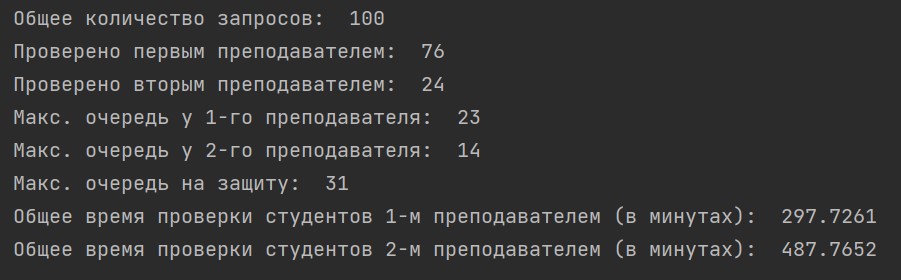
****

Рисунок 2: работа программы при вероятности 80%

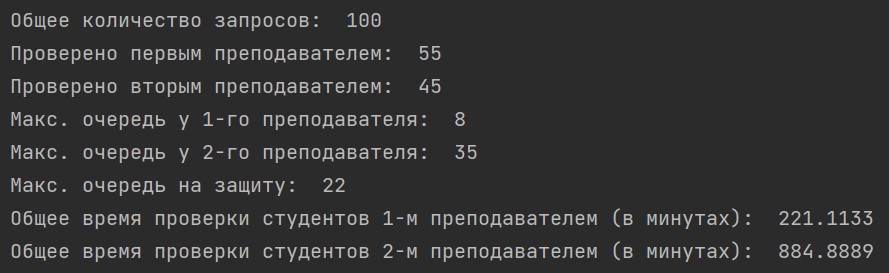


Рисунок 3: работа программы при вероятности 50%

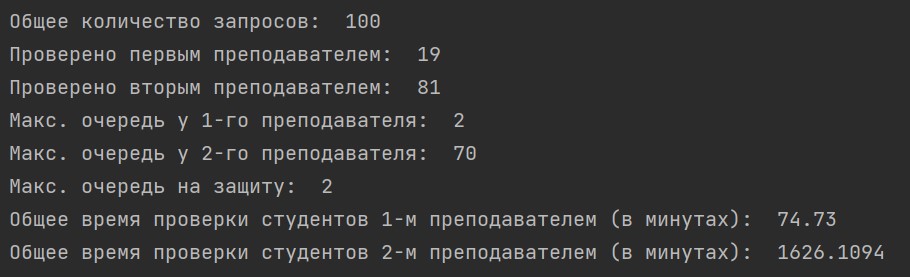


Рисунок 4: работа программы при вероятности 20%