**Назначение работы:** организацияпрограммных запросов к базе данных файла о событиях.

**Теория.**

См. лекцию 12.

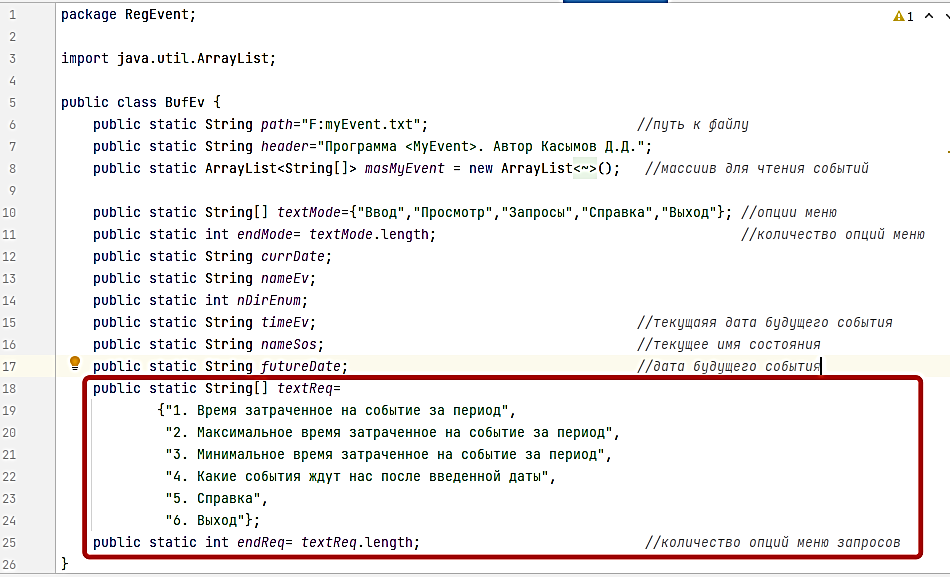
Наша база данных MyEvent имеет нестандартную структуру, поэтому использовать язык запросов невозможно. Реализуем механизм обработки наших данных из файла. Для простоты ограничимся запросами из четырех вариантов:

1. Время затраченное на событие за период.
2. Максимальное время затраченное на событие за период.
3. Минимальное время затраченное на событие за период.
4. Какие события ждут нас после введенной даты.

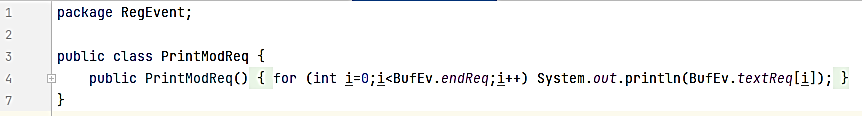
**Практика.**

Открыть свой проект **11\_MyEvent** и переименовать в **12\_ MyEvent.**

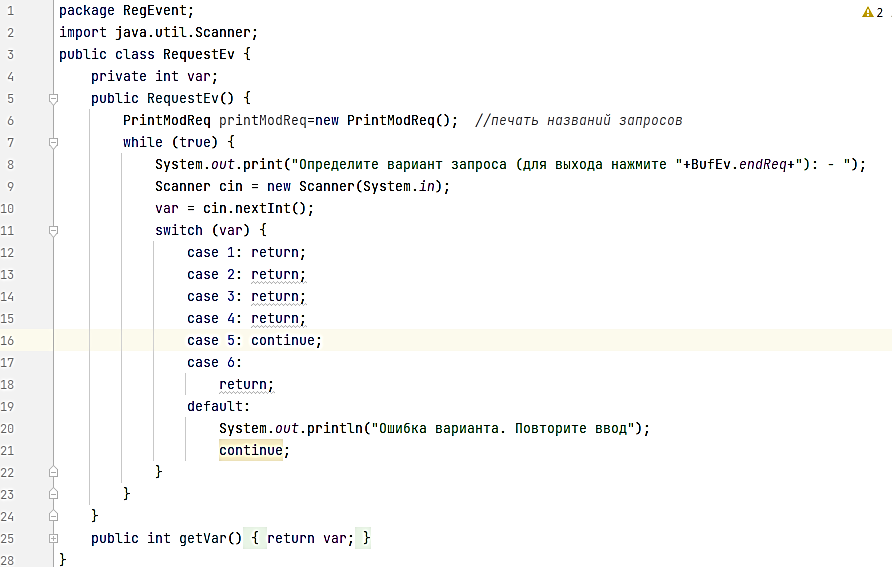
Изменим класс *BufEv:*



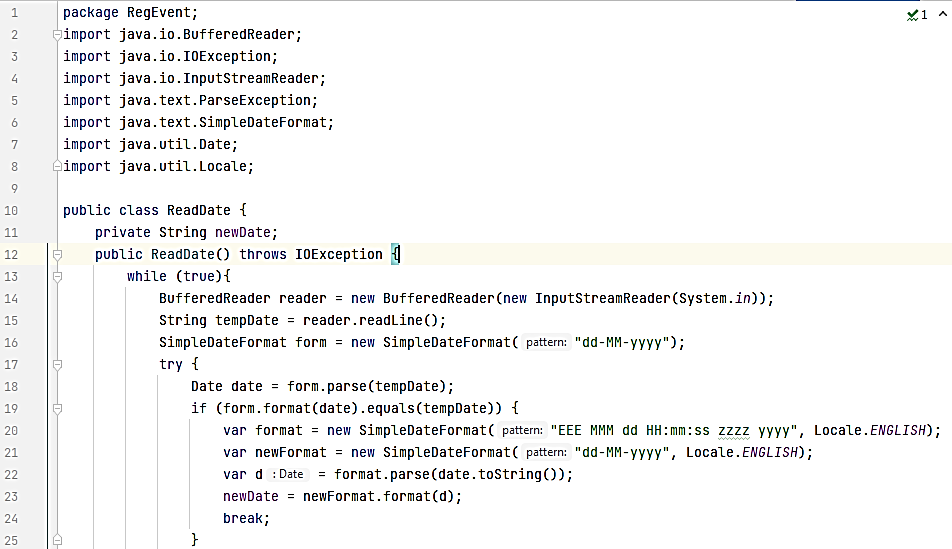
Создадим новый класс PrintModReq для вывода на экран списка названий запросов.

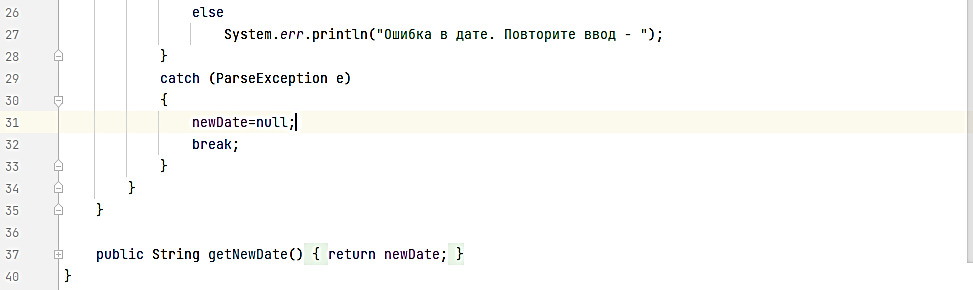


Далее откроем новый класс для ввода и контроля номера запроса:

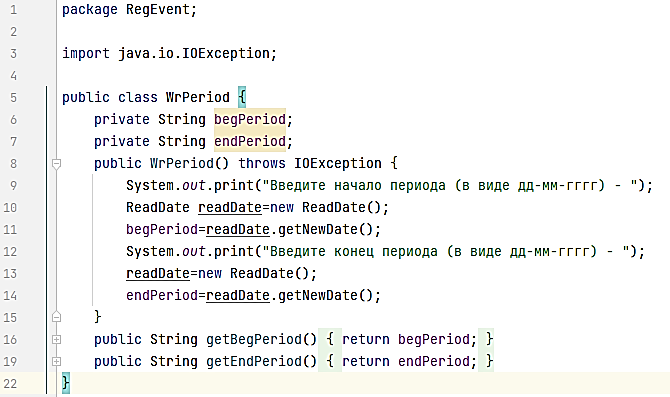


Далее реализуем программу для выполнения запроса по его номеру. Но кроме номера запроса потребуются начало периода, конец периода и название события. Поэтому организуем ввод этих данных. Это может выполнить один класс ReadDate

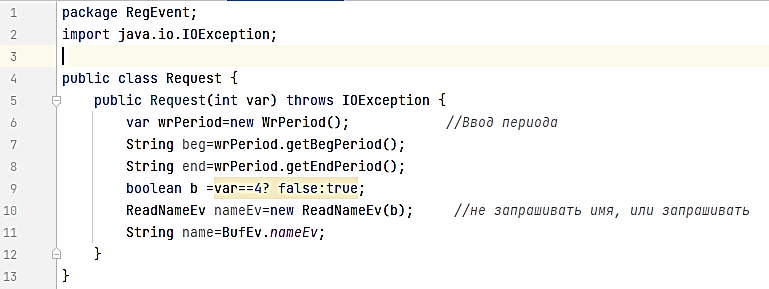




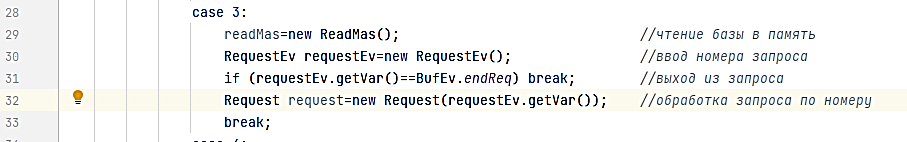
Класс для ввода даты реализован, следовательно, можно разработать класс для ввода периода:



Ввод исходных данных для запроса: номер запроса, период и имя события реализовано в классе Request

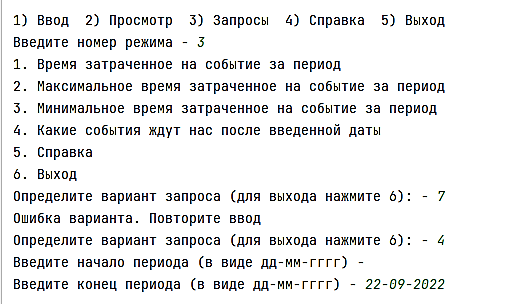


Осталось добавить код в класс *Main*:

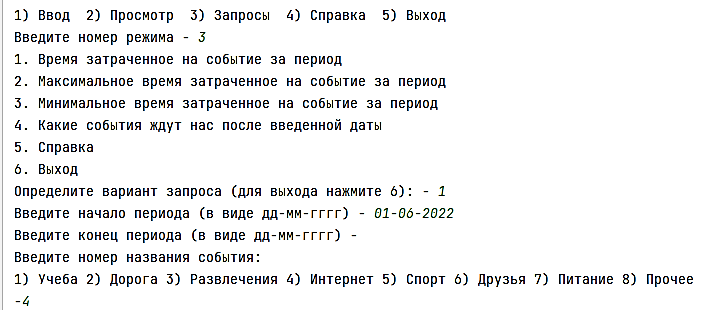


Проверить работу программы.

Пример 1 работы программы:



Пример 2 работы программы:



**Как сдавать работу?**

1. Показать папку с именем Java-Bakay.
2. Открыть проект с именем MyEvent.
3. Прочитать и выдать файл на экран
4. Ввести данные, как в примере 1 и 2
5. Устно объяснить назначение каждого оператора программы.