**Назначение работы:** для организации справки реализовать принцип ООП - наследование.

**Теория.**

См. лекцию 14.

**Наследование –** один из принципов объектно-ориентированного программирования (ООП), который дает возможность создавать новый класс, дочерний к родительскому классу. При этом сохраняются все свойства и методы родительского класса, но при этом можно вводить новые свойства и методы, а также заменять родительские.

**Практика.**

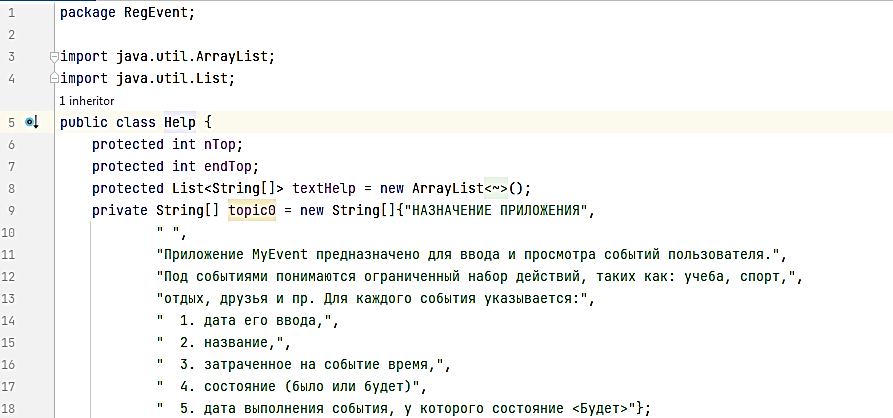
Открыть свой проект **13\_MyEvent** и переименовать в **14\_ MyEvent.**

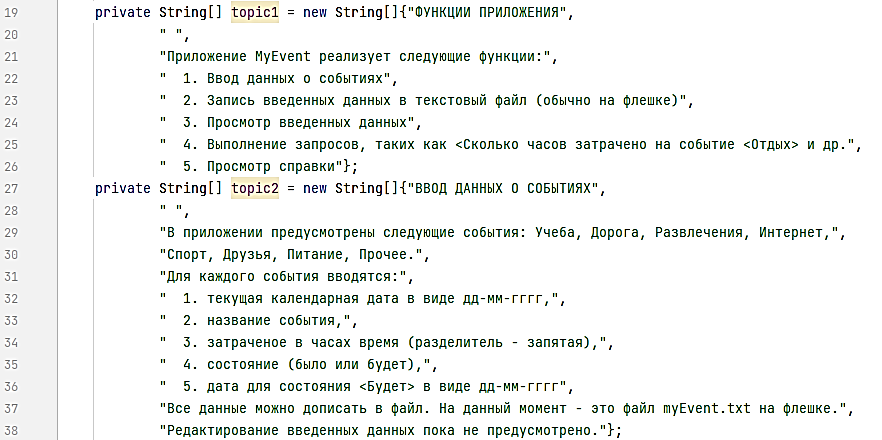
Справка должна выдаваться в соответствии с текущим контекстом. Вся полностью или отдельный ее топик. Т.е. в соответствии с этим требованием справка может быть разной. Для того что бы стало возможным воспользуемся принципом ООП – наследование.

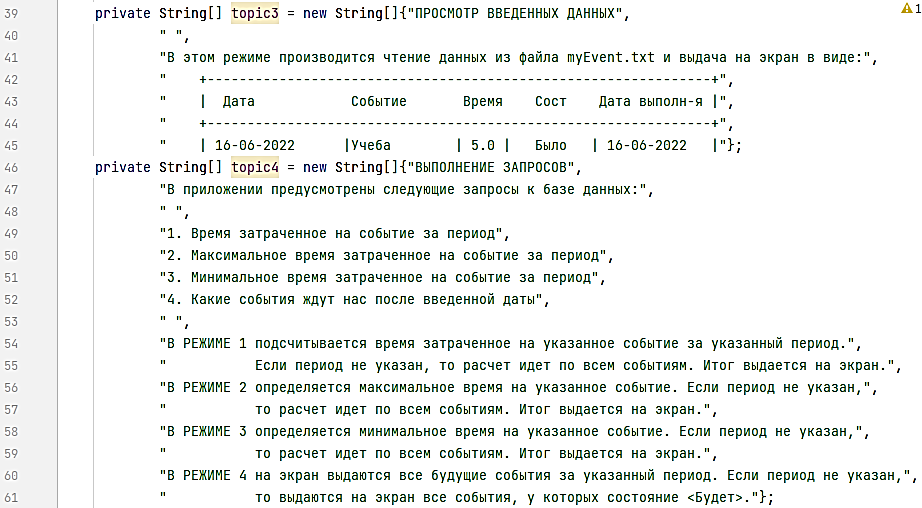
У нас будет родительский класс *Help* и дочерний класс *PrintHelp.*

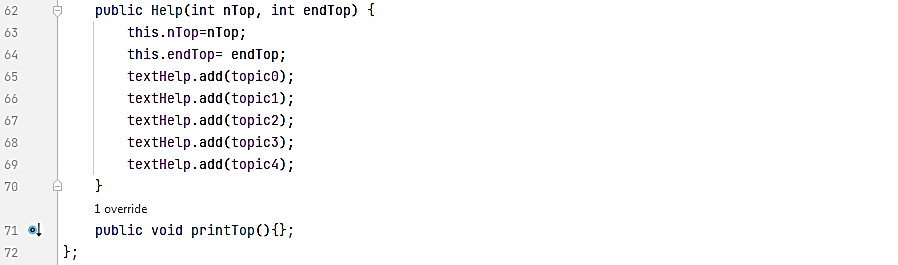
В родительском классе будет текст топиков (разделов) справки и пустой метод для выдачи раздела на экран. В дочернем классе этот метод будет заменен (перезагружен).

Класс *Help*:

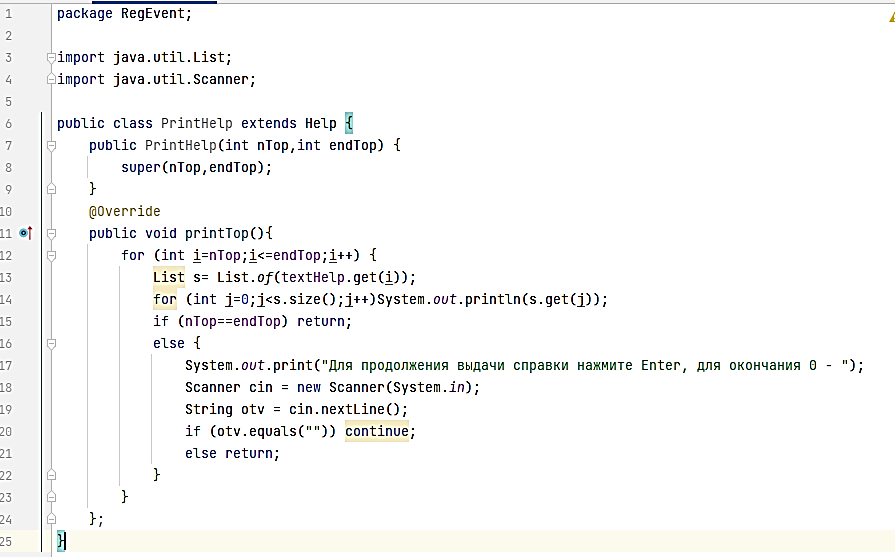




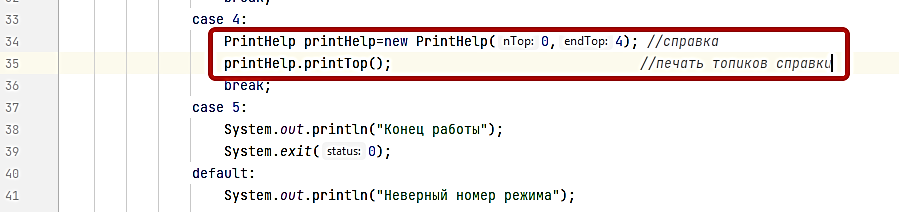




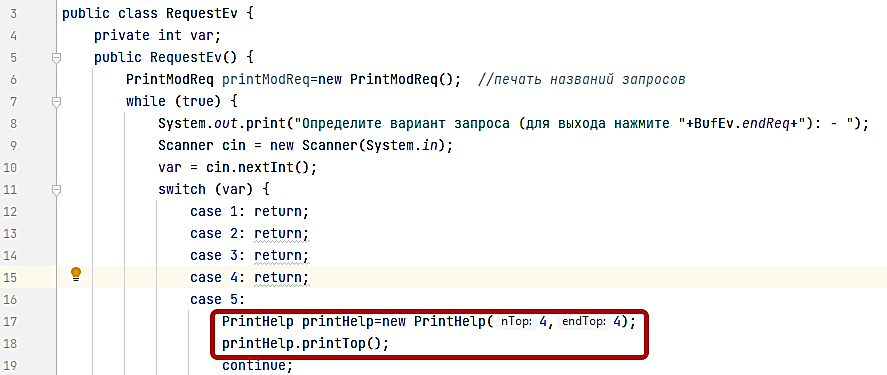
Класс наследник *PrintHelp*:



Измените код класса *Main:*

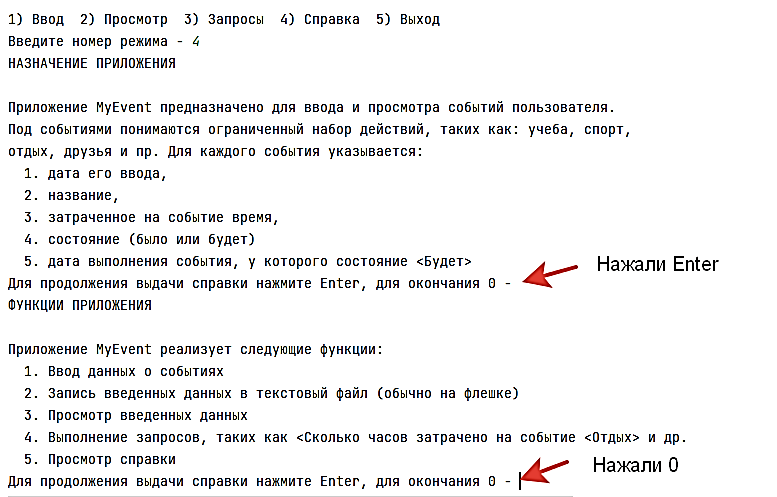


Изменение в меню запросов:



Проверить работу на следующих примерах:

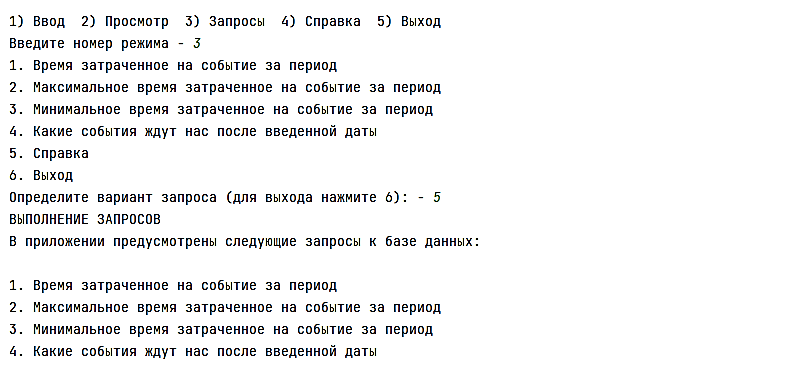
Пример 1.

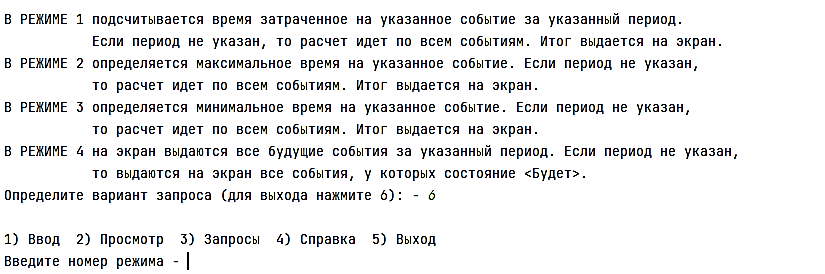


Пример 2.

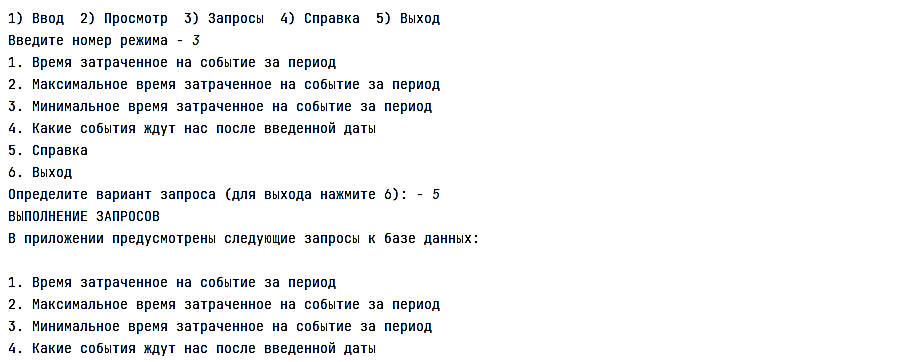
Вызвать справку и выдать ее до конца, нажимая *Enter*

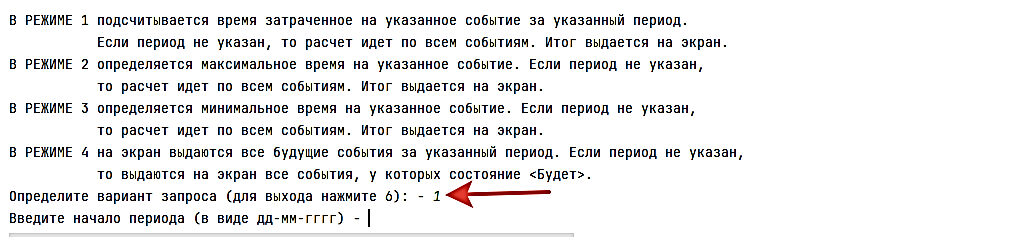
Пример 3.





Пример 4.





Показать папку с именем Java-Bakay.

1. Открыть проект с именем MyEvent.
2. Прочитать и выдать файл на экран
3. Выполнить примеры 1-4
4. Устно объяснить назначение каждого оператора программы.