Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 9(А).** **Пакеты и импорт, наследование, Модификаторы видимости в Kotlin.**»

Выполнил: Гордейчик Степан Алексеевич

Группа: ПР-21

Преподаватель: Мирошниченко Г. В.

2023

Добавить классы-наследники с дополнительным своим конструктором и функциями;

**Вариант 7.** Класс «Собака»

* Первый вариант класса-наследника (Служебная собака).

**Входные данные:**

name – кличка собаки (String) – Основной класс

breed – порода собаки (String) – Основной класс

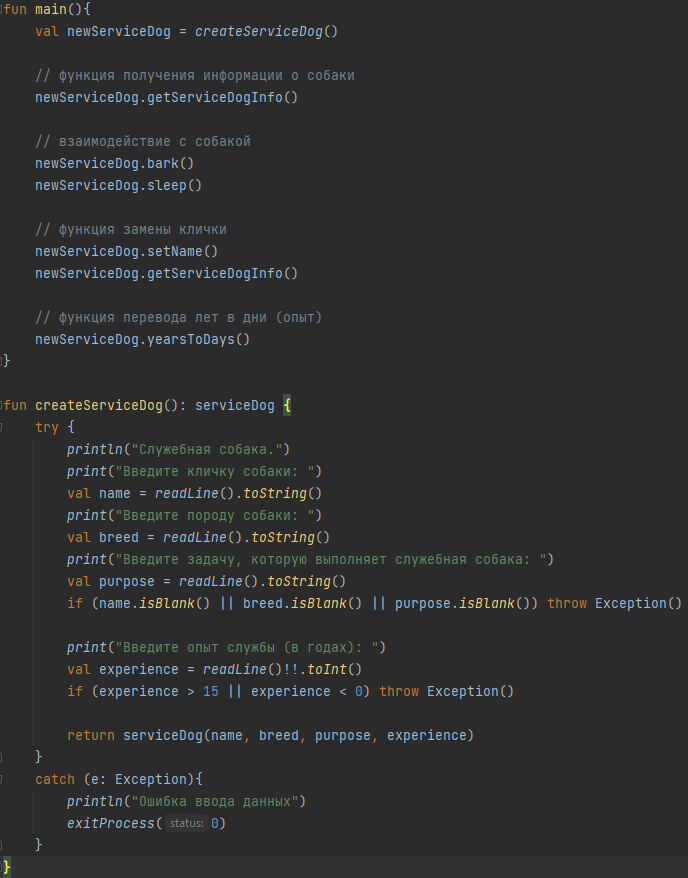
purpose – задача, которую выполняет служебная собака (String) – Класс-наследник

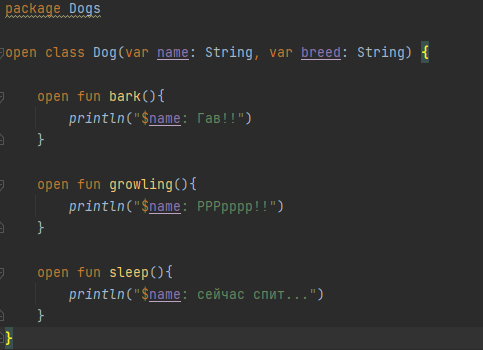
experience – опыт службы собаки (в годах) (Int) – Класс-наследник

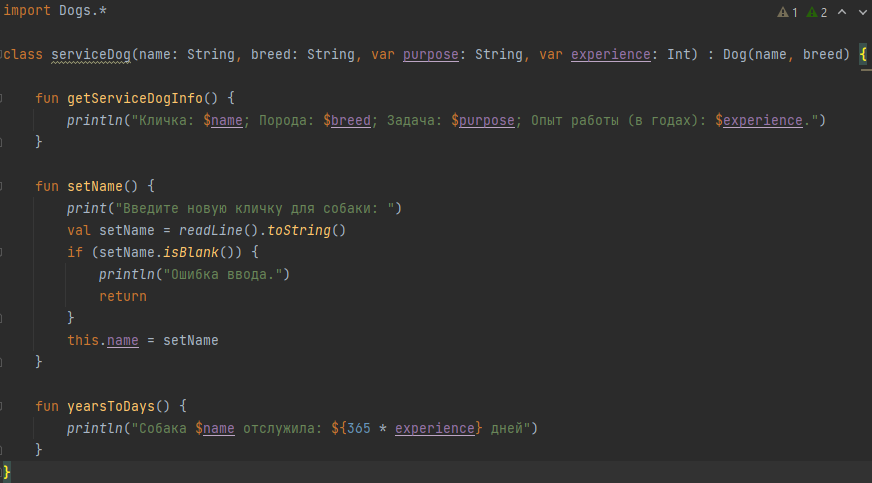
**Выходные данные:**

Текстовые сообщения.

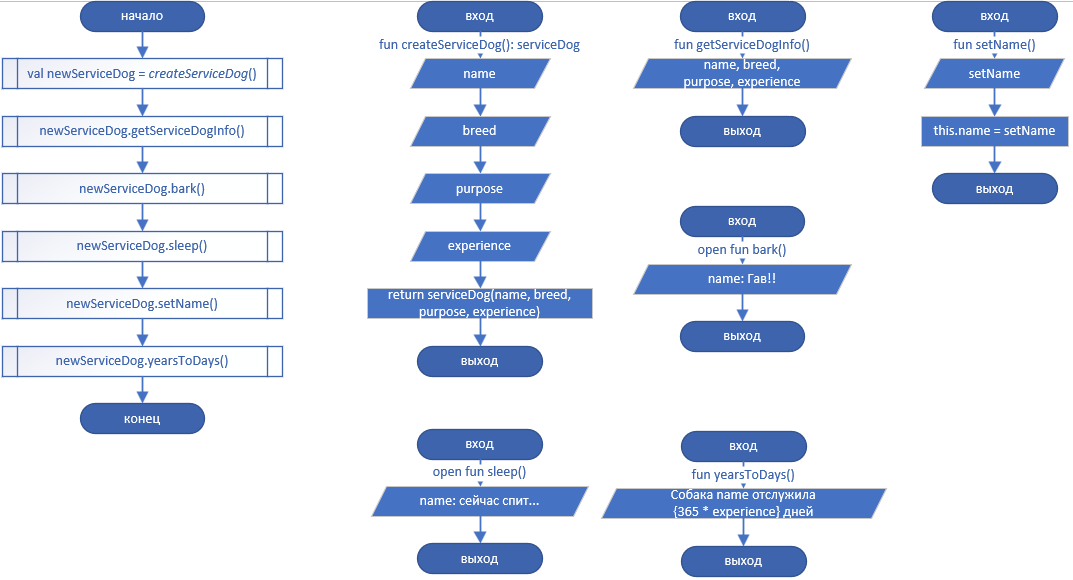
**Листинг программы:**

****

****

****

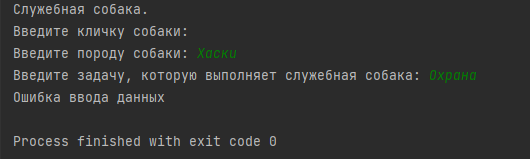
**Блок-схема:**



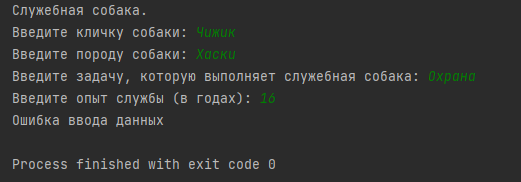
**Тестовые ситуации:**

При вводе начальных данных используется структура try-catch.

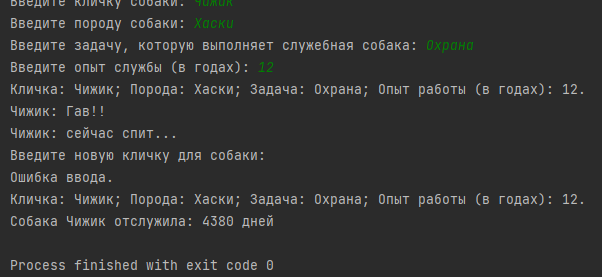
Имеется проверка на пустоту строк при начальном вводе.



Имеется проверка на ограничения опыта службы (от 0 до 15)



Имеется проверка на ввод новой клички (проверка пустоты строки)



* Второй вариант класса-наследника (Спортивная собака).

**Входные данные:**

name – кличка собаки (String) – Основной класс

breed – порода собаки (String) – Основной класс

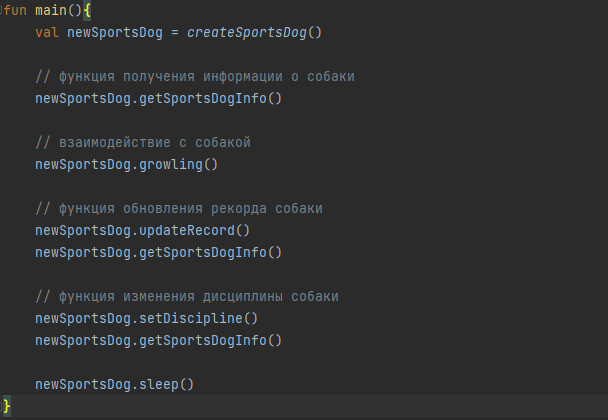
discipline –спортивная дисциплина собаки (String) – Класс-наследник

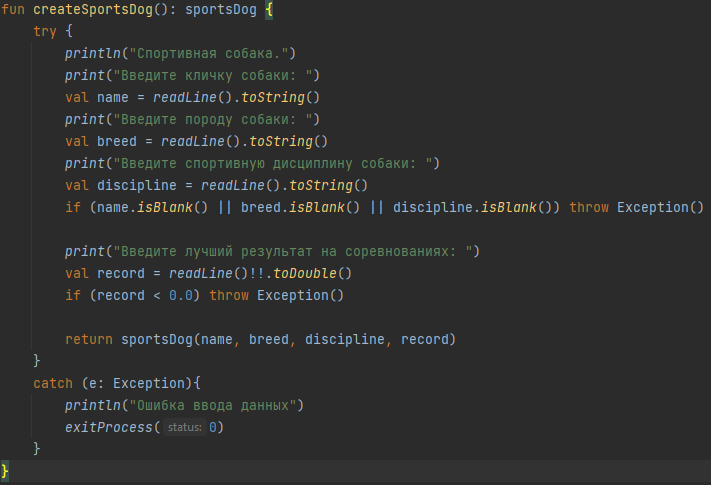
record – лучший результат собаки на соревнованиях (Double) – Класс-наследник

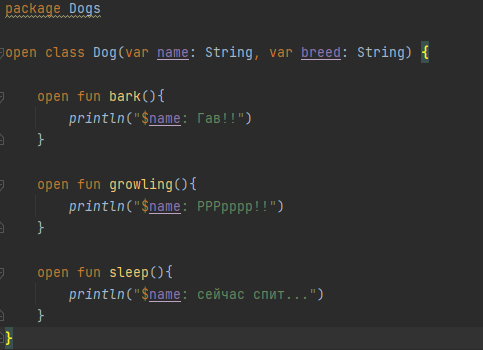
**Выходные данные:**

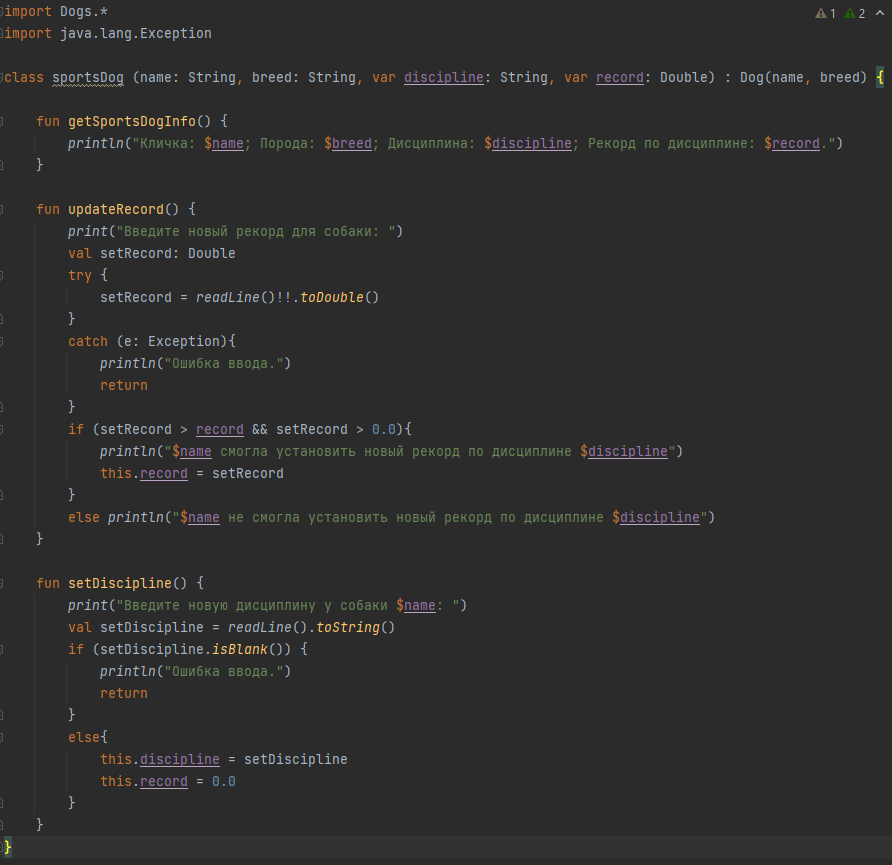
Текстовые сообщения.

**Листинг программы:**

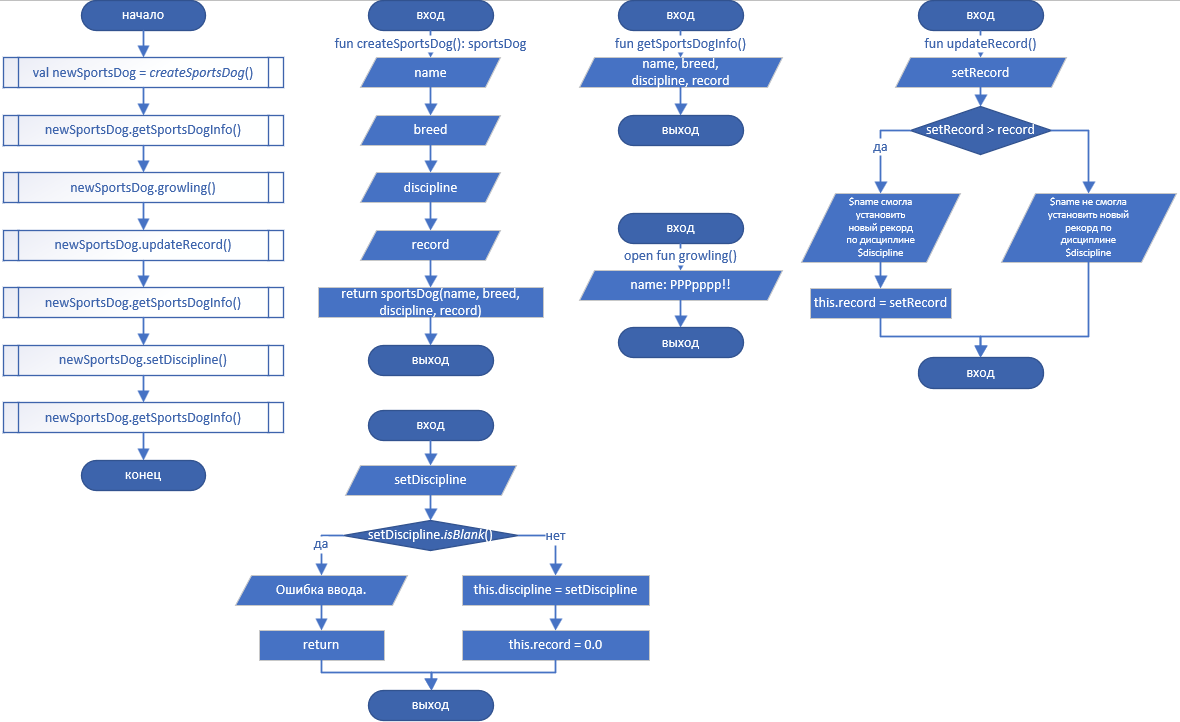




****

****

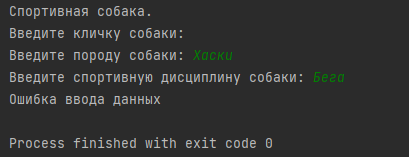
**Блок-схема:**



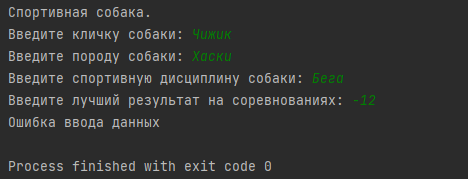
**Тестовые ситуации:**

При вводе начальных данных используется структура try-catch.

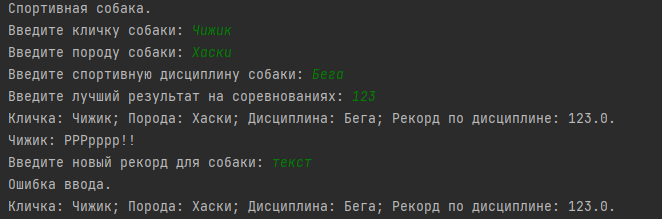
Имеется проверка на пустоту строк при начальном вводе.

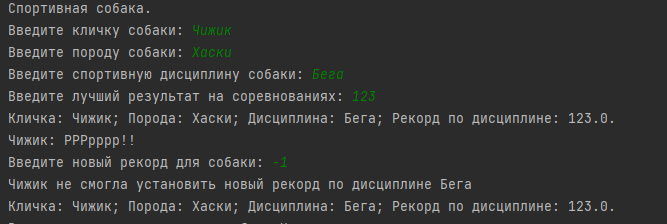


Имеется проверка на запрет ввода отрицательного результата на соревнованиях.

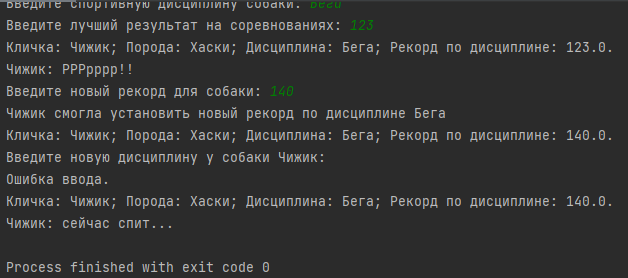


При использовании функции ‘updateRecord()’ используется конструкция try-catch, а также проверка ввода на отрицательное значение.





При использовании функции ‘setDiscipline()’ используется проверка на пустоту строки.



**Вариант 12.** Класс «Кинотеатр»

* Первый вариант класса-наследника (Фильм кинотеатра).

**Входные данные:**

name – название кинотеатра (String) – Основной класс

ageCinema – возраст кинотеатра (Int) – Основной класс

title – название фильма (String) – Класс-наследник

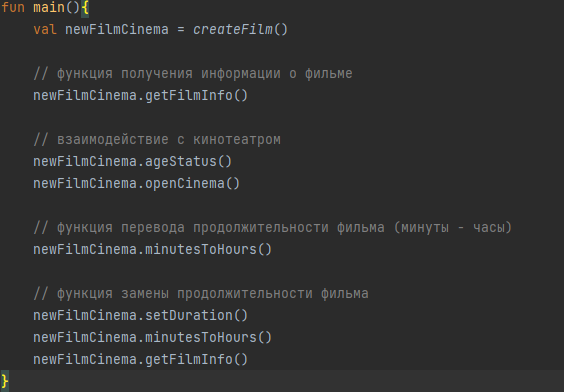
duration –продолжительность фильма (в минутах) (Int) – Класс-наследник

genre – жанр фильма (String) – Класс-наследник

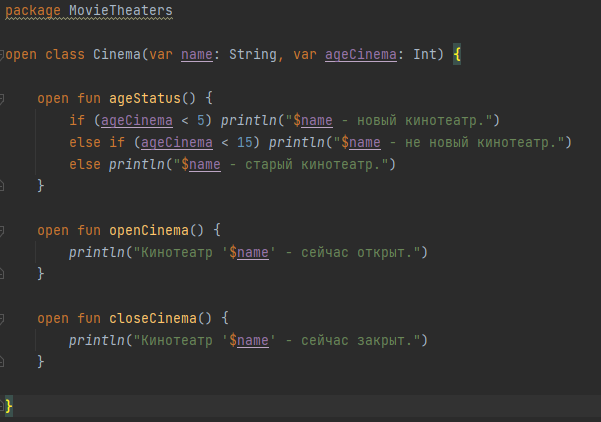
**Выходные данные:**

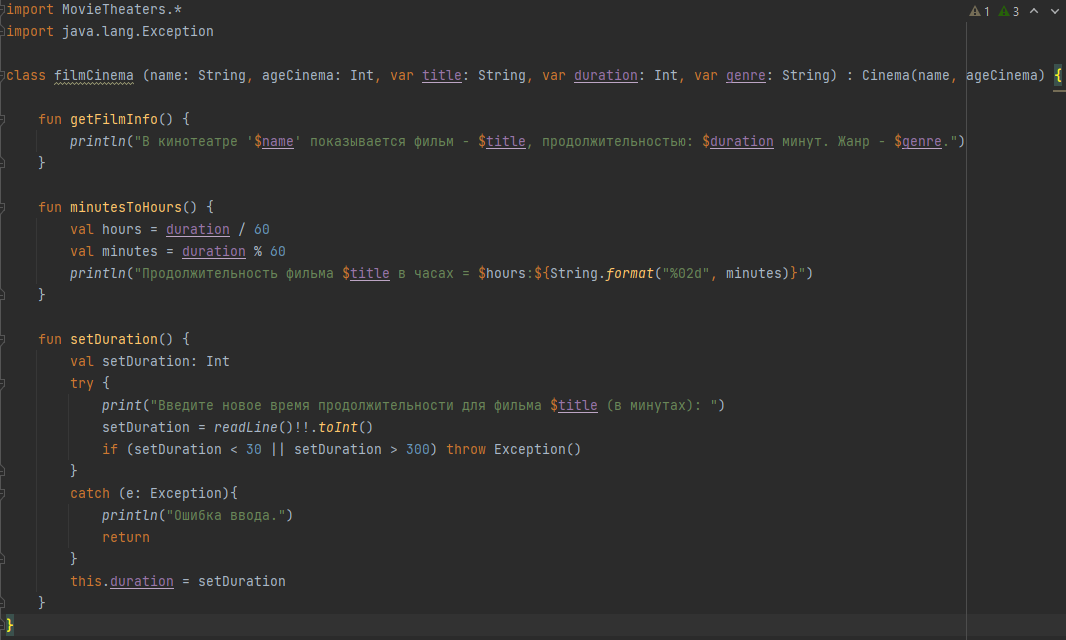
Текстовые сообщения.

**Листинг программы:**

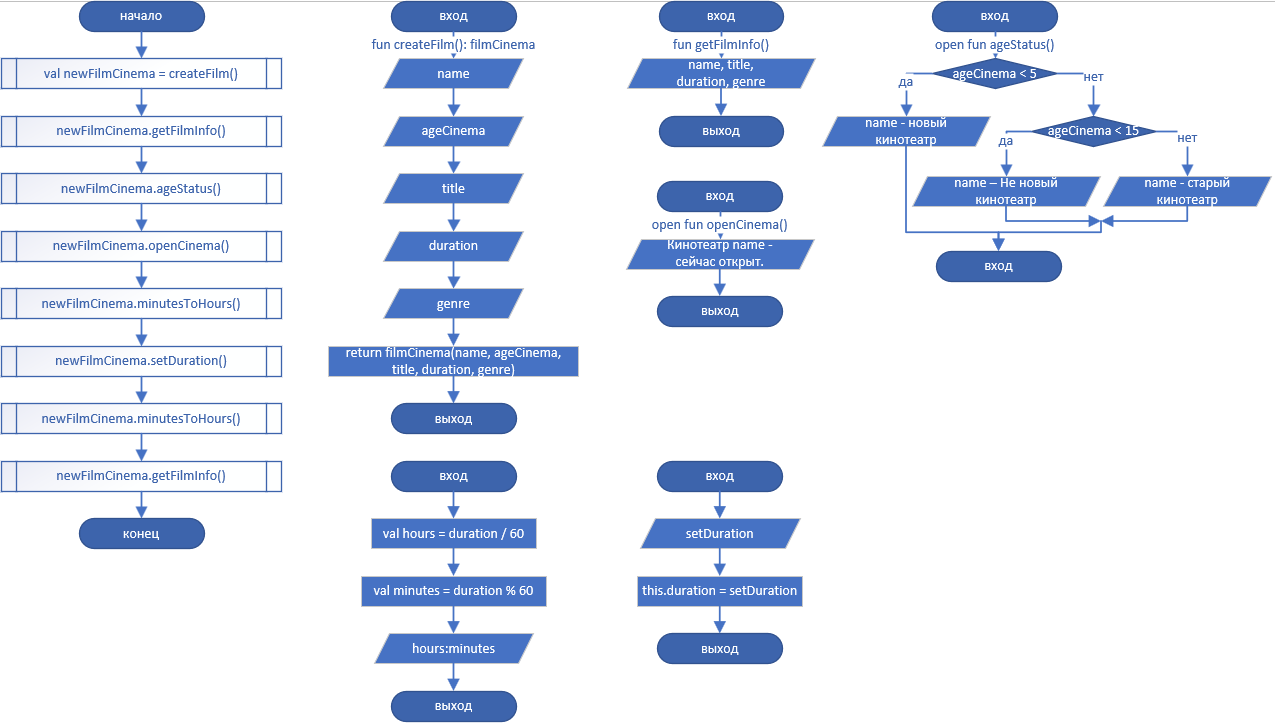








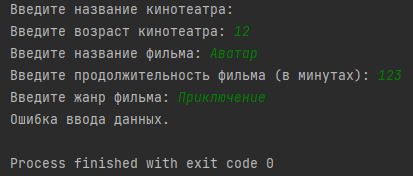
**Блок-схема:**



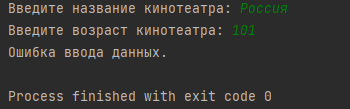
**Тестовые ситуации:**

При вводе начальных данных используется структура try-catch.

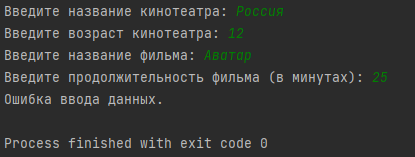
Имеется проверка на пустоту строк при начальном вводе.



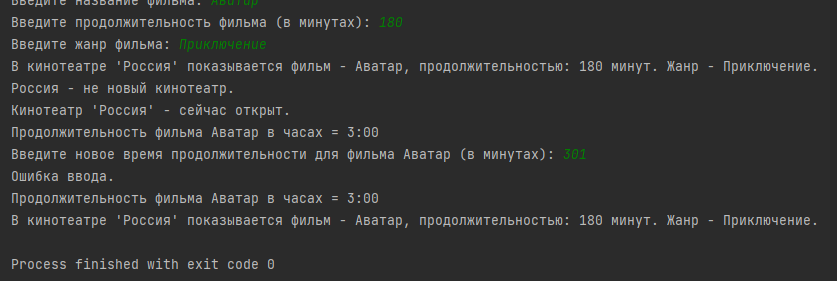
Имеется проверка на ограничение ввода возраста кинотеатра (от 0 до 100).



Имеется проверка на ограничение ввода продолжительности фильма (от 30 до 300).



При использовании функции ‘setDuration()’ используется конструкция try-catch, а также проверка ввода на допустимые значения (от 30 до 300).



* Второй вариант класса-наследника (Сеанс кинотеатра).

**Входные данные:**

name – название кинотеатра (String) – Основной класс

ageCinema – возраст кинотеатра (Int) – Основной класс

film – название сеанса (String) – Класс-наследник

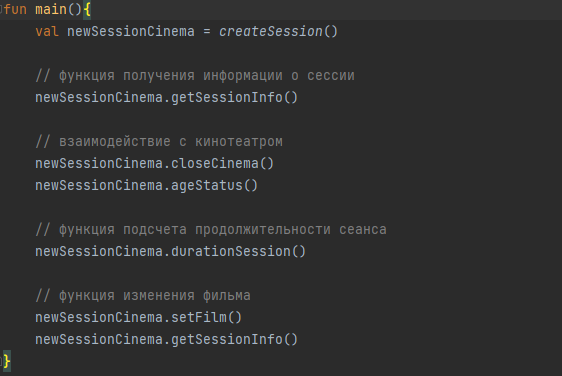
startTime – время начала сеанса (час:мин) (LocalTime) – Класс-наследник

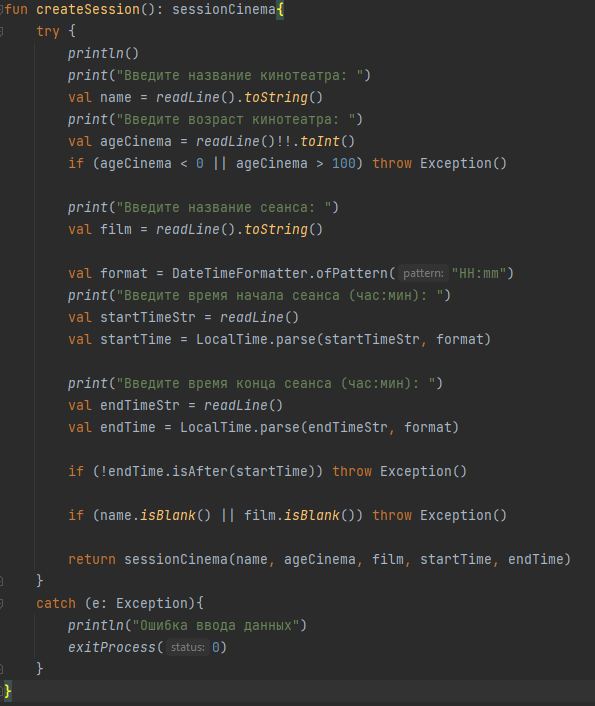
endTime – время конца сеанса (час:мин) (LocalTime) – Класс-наследник

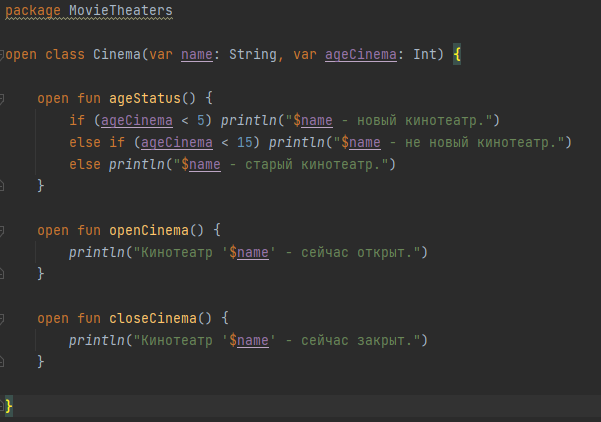
**Выходные данные:**

Текстовые сообщения.

**Листинг программы:**

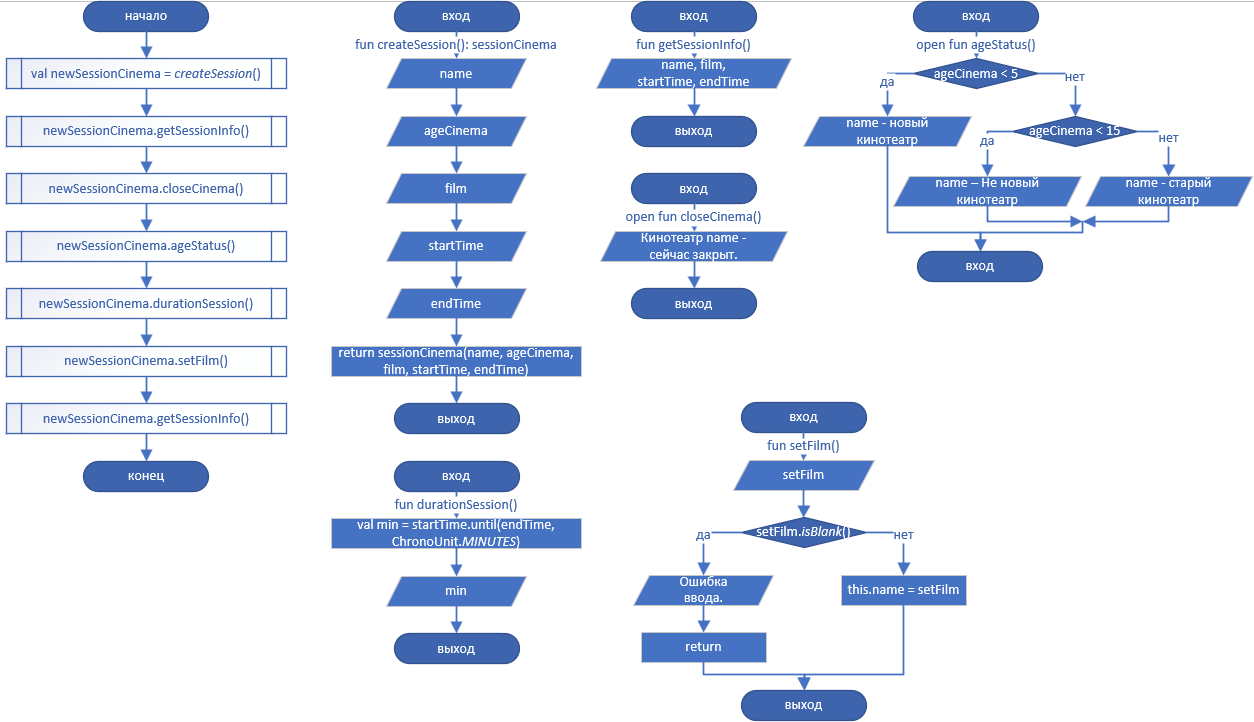








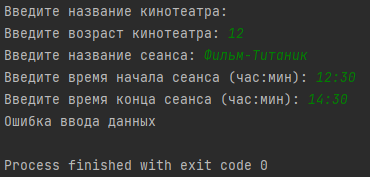
**Блок-схема:**



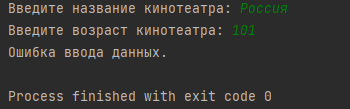
**Тестовые ситуации:**

При вводе начальных данных используется структура try-catch.

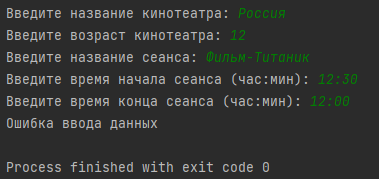
Имеется проверка на пустоту строк при начальном вводе.



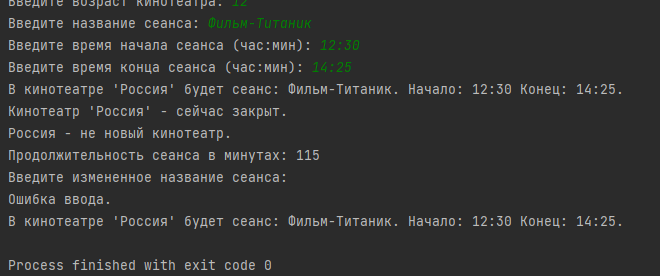
Имеется проверка на ограничение ввода возраста кинотеатра (от 0 до 100).



Имеется проверка, что начальная дата не превышает конечную.



При использовании функции ‘setFilm()’ используется проверка на пустоту строки.



Вывод: я научился использовать пакеты, импорт и наследование в Kotlin.