Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическая № 10**»

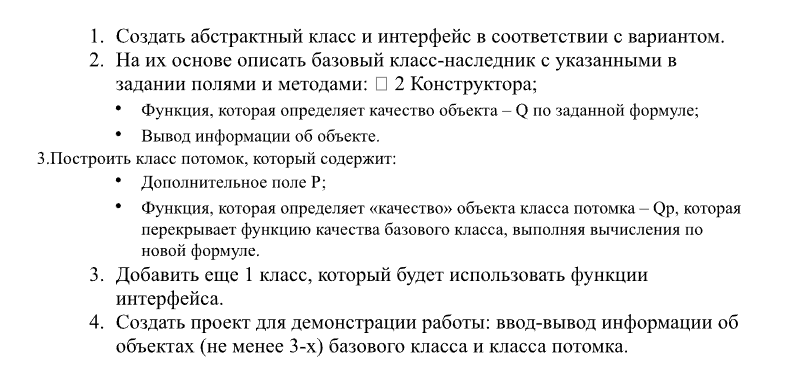
Выполнил: Карамов Д.Э

Группа: ПР-24

Преподаватель: Мирошниченко Г.В

2025

**Задание:**

****

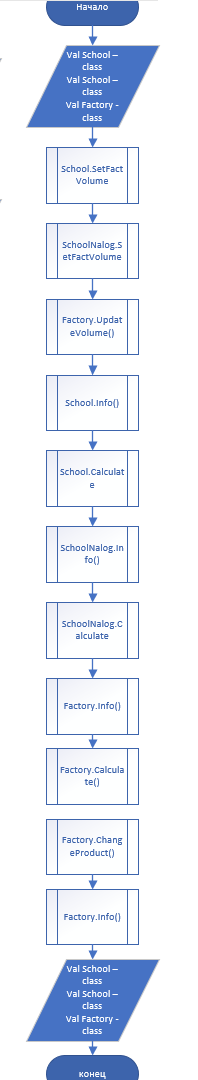
**Входные и выходные данные**

Val School – class – класс наследник

Val SchoolNalog – class – класс потомок

Val Factory – class с интерфейсом

**Блок-схема**

****

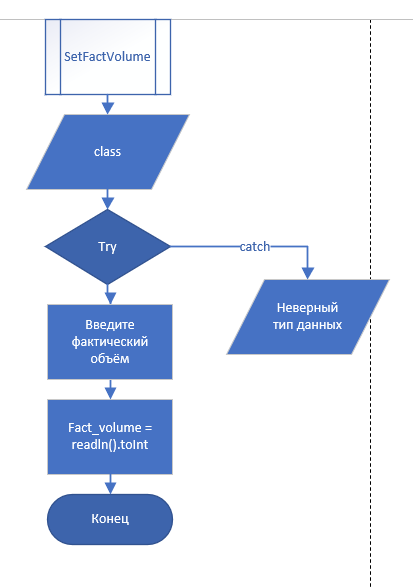
**Подпрограмма SetFactVolume()**

**Входные и выходные данные**

Val School – class – класс наследник

Val SchoolNalog – class – класс потомок

**Блок схема**

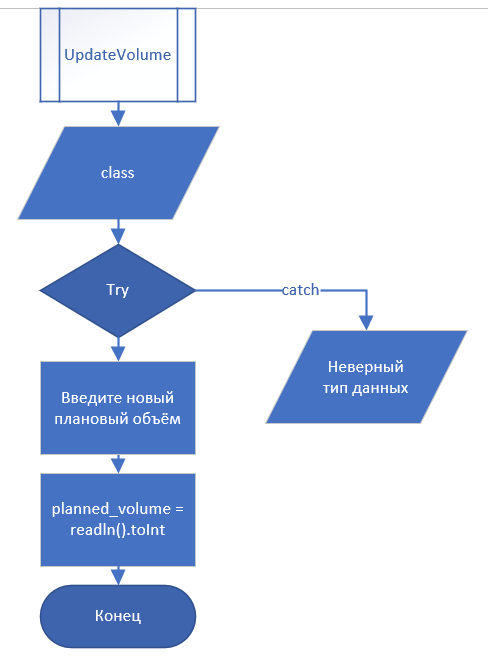


**Подпрограмма UpdateVolume()**

**Входные и выходные данные**

Val Factory – class с интерфейсом

**Блок схема**

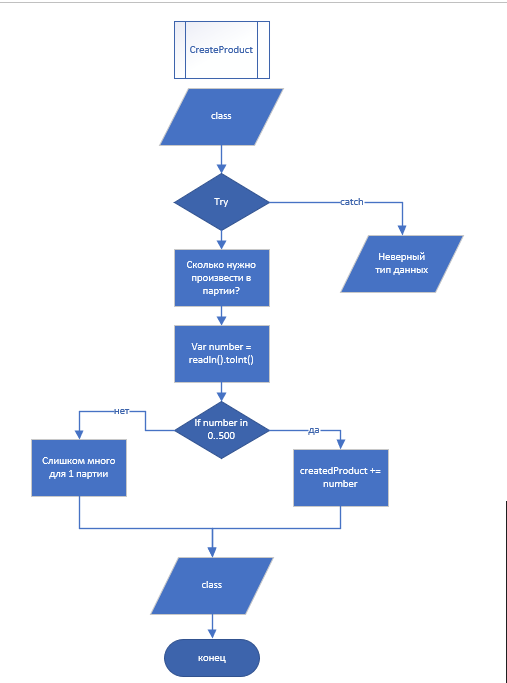


**Подпрограмма CreateProduct()**

**Входные и выходные данные**

Val Factory – class с интерфейсом

**Блок схема**

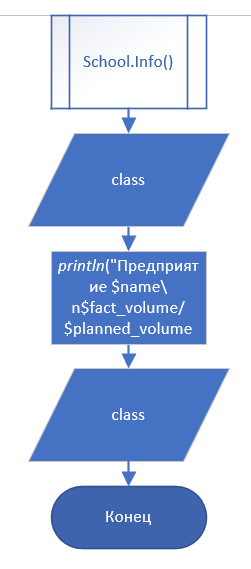


**Подпрограмма School.Info()**

**Входные и выходные данные**

Val School – class – класс наследник

**Блок схема**

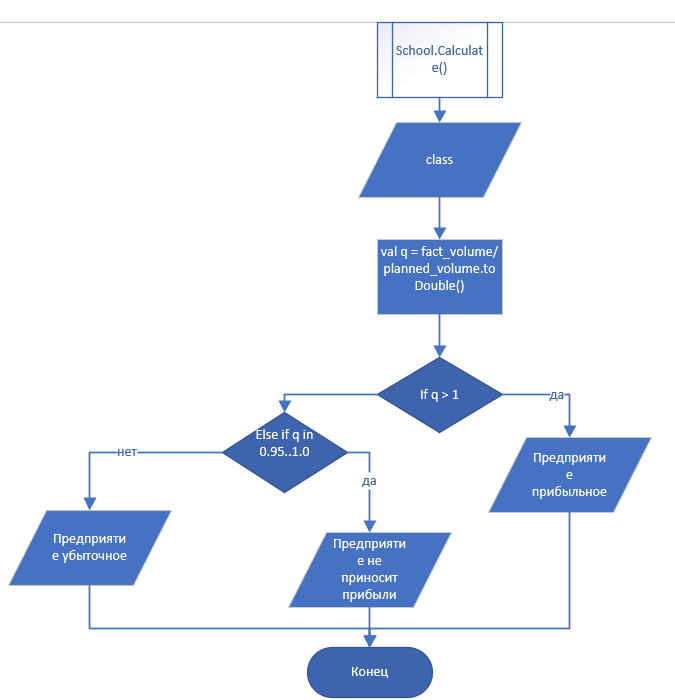


**Подпрограмма School.Calculate()**

**Входные и выходные данные**

Val School – class – класс наследник

**Блок схема**

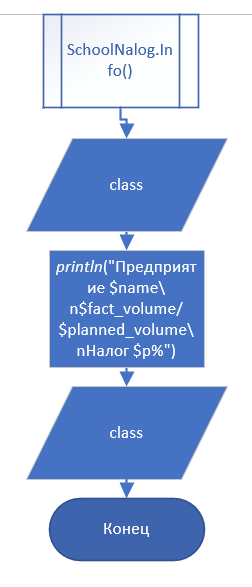


**Подпрограмма SchoolNalog.Info()**

**Входные и выходные данные**

Val SchoolNalog – class – класс потомок

**Блок схема**

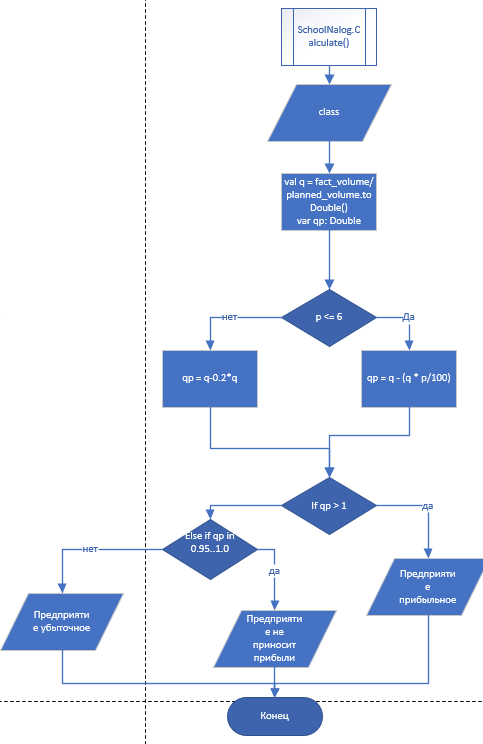
****

**Подпрограмма SchoolNalog.Calculate()**

**Входные и выходные данные**

Val SchoolNalog – class – класс потомок

**Блок схема**

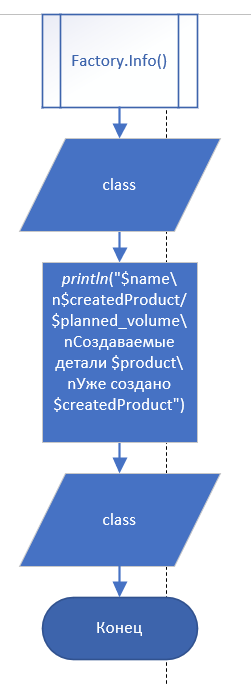
****

**Подпрограмма Factory.Info()**

**Входные и выходные данные**

Val Factory – class с интерфейсом

**Блок схема**

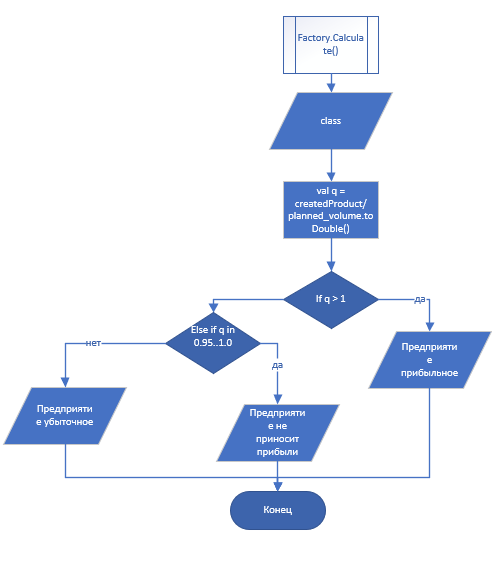
****

**Подпрограмма Factory.Calculate()**

**Входные и выходные данные**

Val Factory – class с интерфейсом

**Блок схема**



**Подпрограмма Factory.ChangeProduct ()**

**Входные и выходные данные**

Val Factory – class с интерфейсом

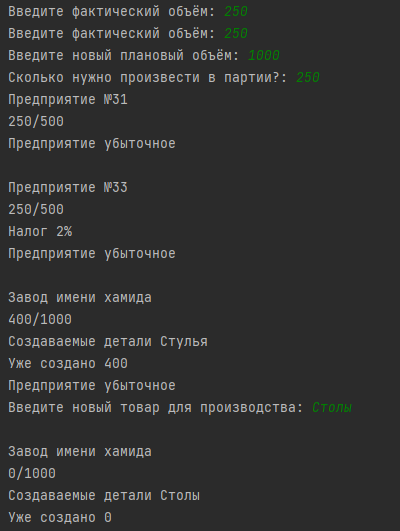
**Блок схема**



Листинг основной программы

fun main()  
{  
 val School = School("№31", 500)  
 val SchoolNalog = SchoolNalog("№33", 500)  
 val Factory = Factory("Завод имени хамида", 1000, "Стулья", 150)  
 School.SetFactVolume() //Ввод фактического объёма для школы  
 SchoolNalog.SetFactVolume() //Ввод фактического объёма для школы, которая платит налог  
 Factory.UpdateVolume() //Ввод планового объёма для завода  
 Factory.CreateProduct() //Создание продукта на заводе (тоже самое что и фактический объём)  
 School.Info()  
 School.Calculate()  
 *println*()  
 SchoolNalog.Info()  
 SchoolNalog.Calculate()  
 *println*()  
 Factory.Info()  
 Factory.Calculate()  
 Factory.ChangeProduct() //Изменение создаваемого на заводе продукта  
 *println*()  
 Factory.Info()  
}

Результат



Тестовые ситуации





Вывод

Я научился создавать абстрактные классы и интерфейсы