Министерство образования и молодежной политики Свердловской области



ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по программе «**Практическое занятие 8**»

Выполнил: Горовой В.С

Группа: ПР-22

Преподаватель: Мирошниченко Г.В

2024

**Задание:**

****

****

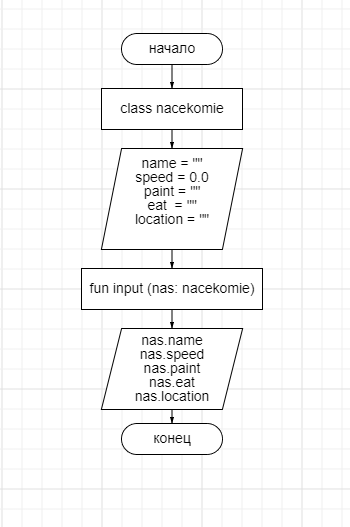
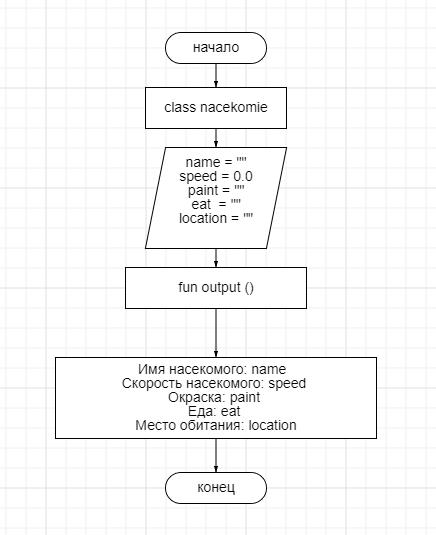
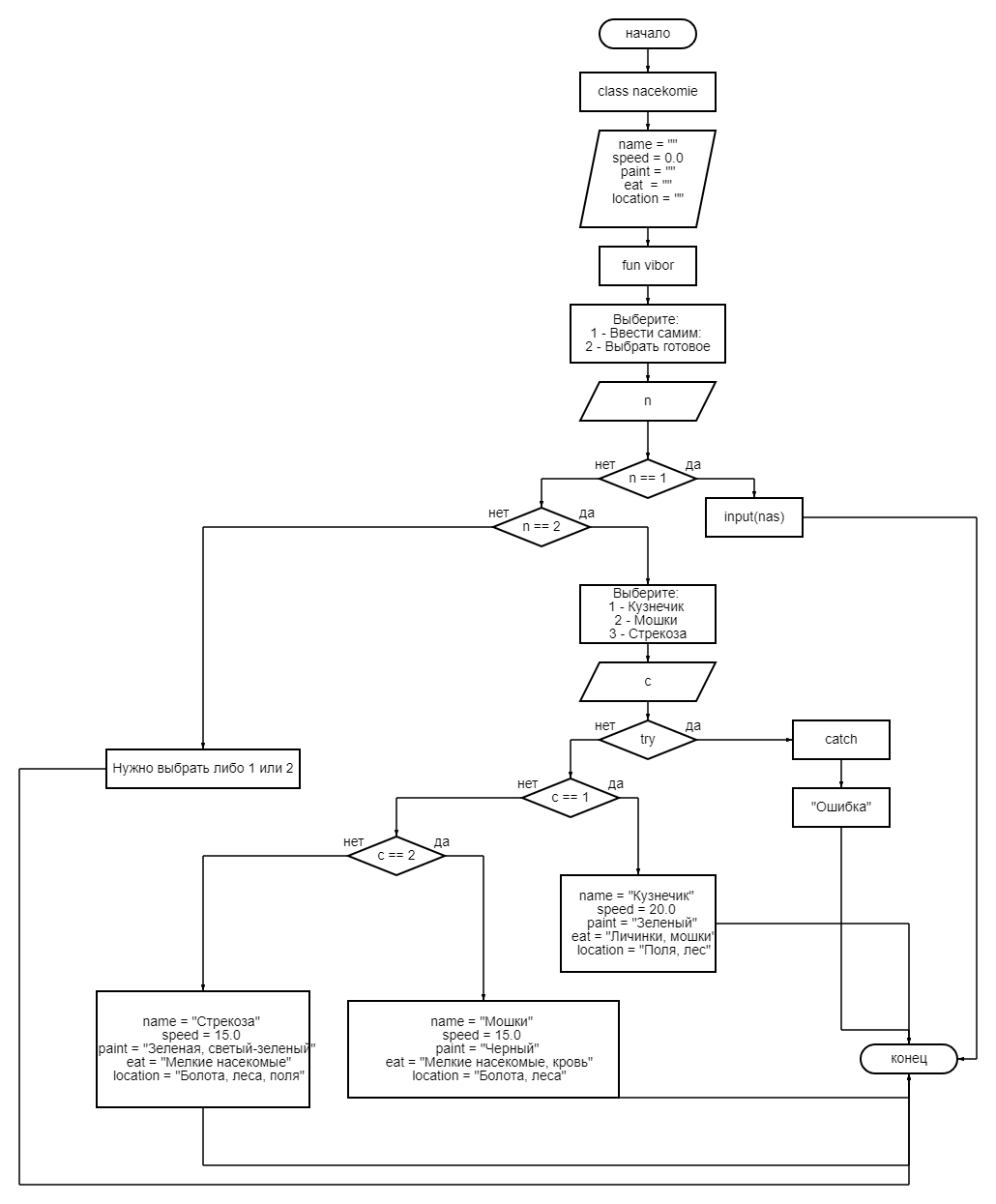
**Входные и выходные данные**

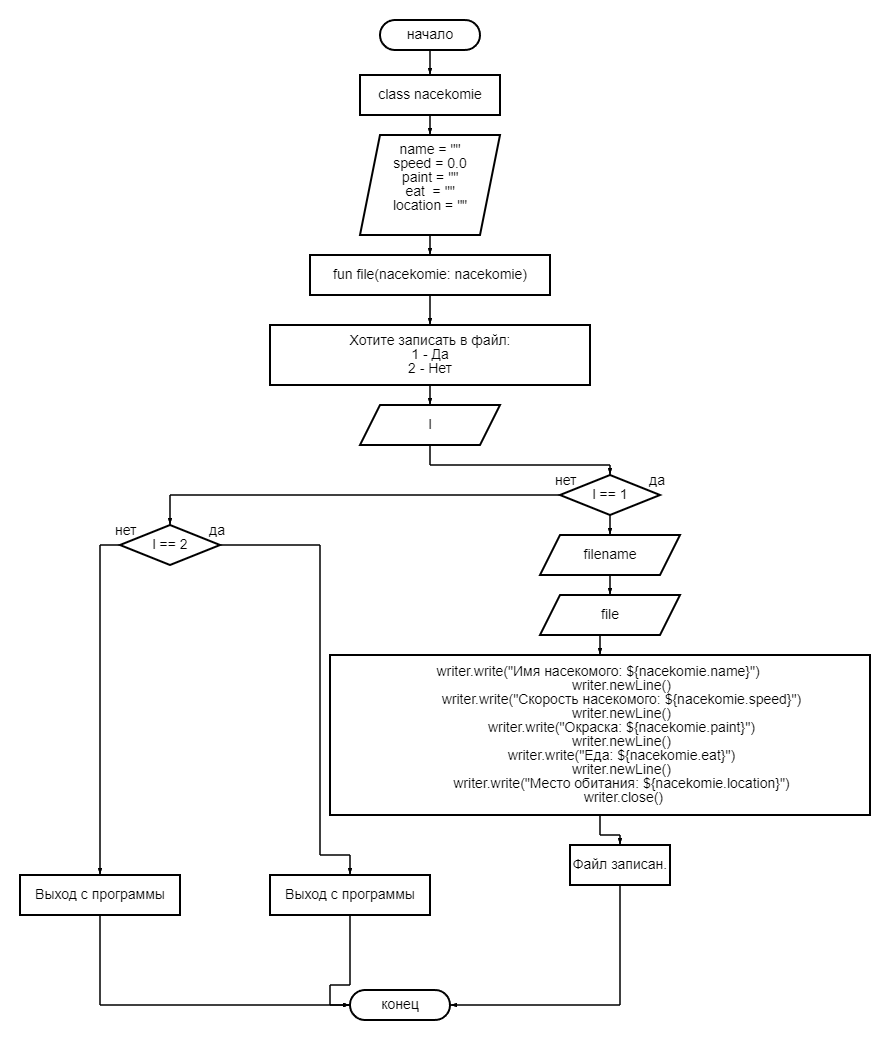
**Вариант 2.  
Класс nacekomie:**

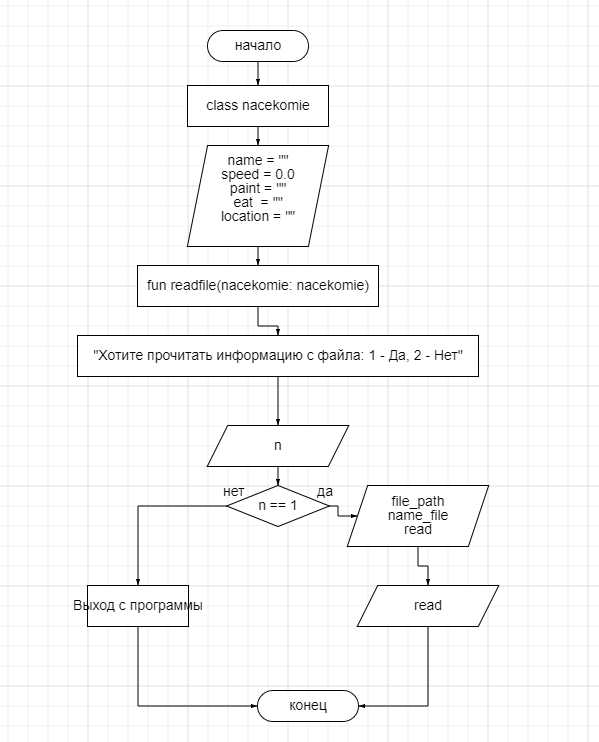
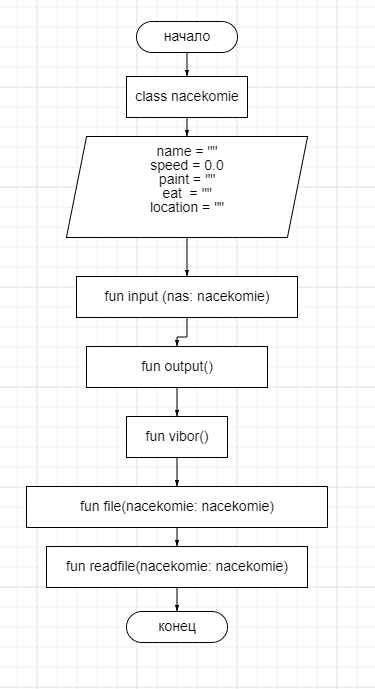
**Входные данные:  
name – имя насекомого, string, строковый тип.  
speed – скорость насекомого, double, дробное число.  
paint – окраска насекомого, string, строковый тип.  
eat – еда, string, строковый тип.  
location – место обитания насекомого, string, строковый тип.  
  
Выходные данные:  
Строка с данными на вывод (name, speed, paint, eat, location)  
  
Main  
  
Входные данные:  
b – переменная для использования класса в проекте.  
Выходные данные:  
Строка с методов класса nacekomie.  
  
Вариант 12.  
Класс cinema:  
  
Входные данные:  
name\_cinema – название кинотеатра, string, строковый тип.  
siance – сеанс, string, строковый тип.  
price – цена билета, double, дробное число.  
zal – зал, string, строковый тип.  
count\_zritelei – количество зрителей, double, дробное число.  
Main  
Входные данные:  
b – переменная для вызова класса  
yes\_or\_no – выбор пользователя, string, строковый тип.  
Выходные данные:  
url – ссылка на сайт с кинотеатрами (если пользователь ввел в yes\_or\_no “Да”), константа.**

**Строка из методов класса cinema**

**Блок-схема**

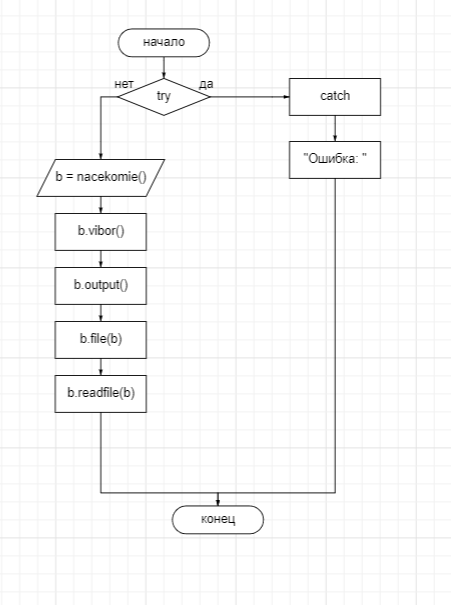
**Вариант 2  
Класс nacekomie  
  
1.   
  
2.**   
  
**3.**

****

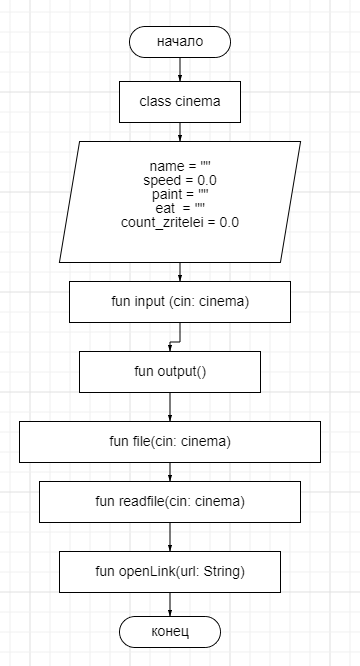
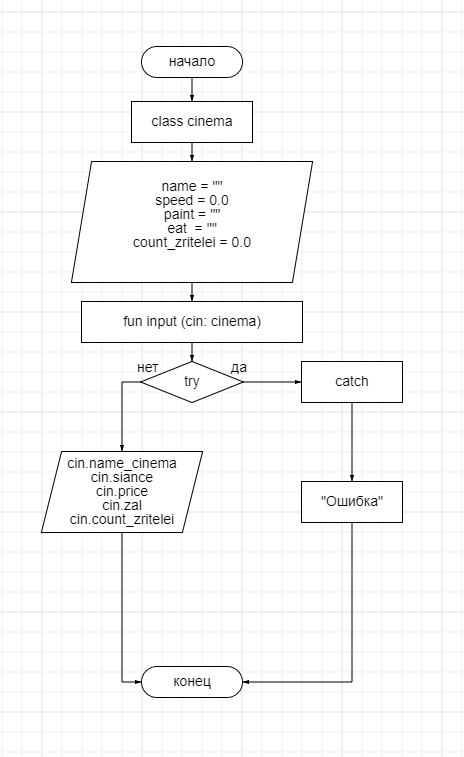
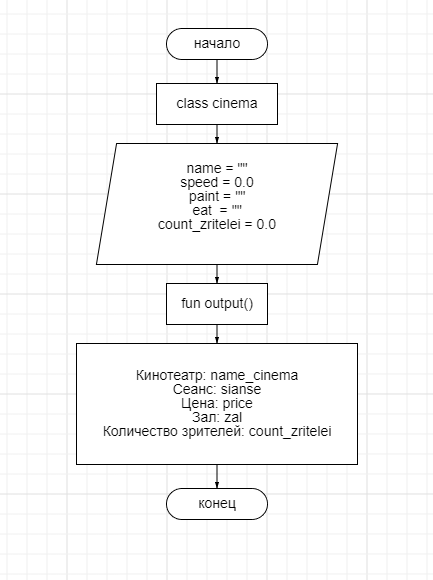
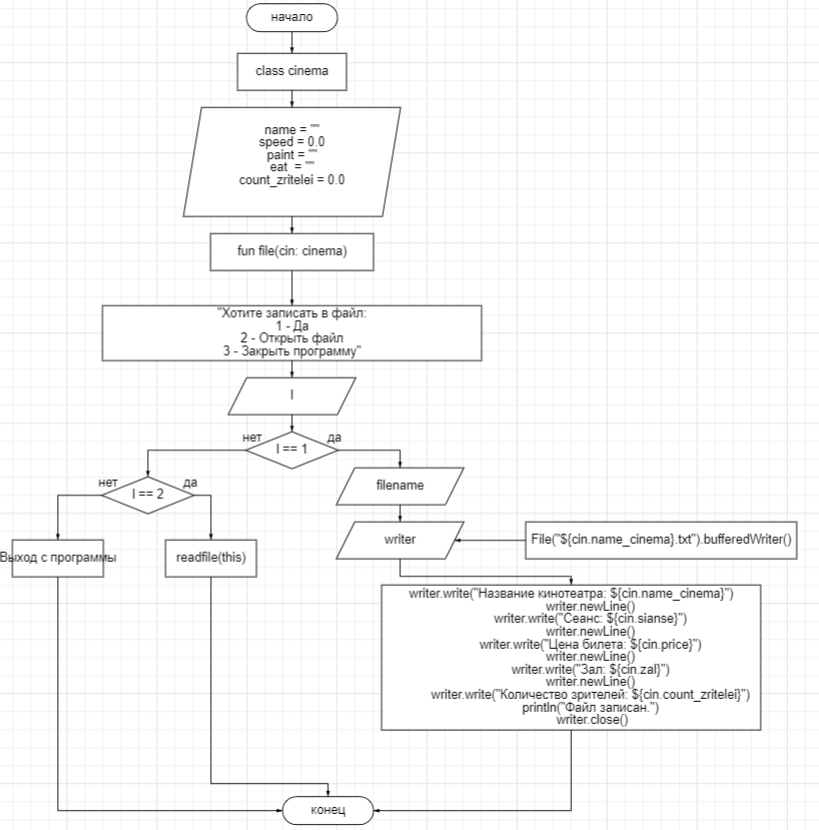
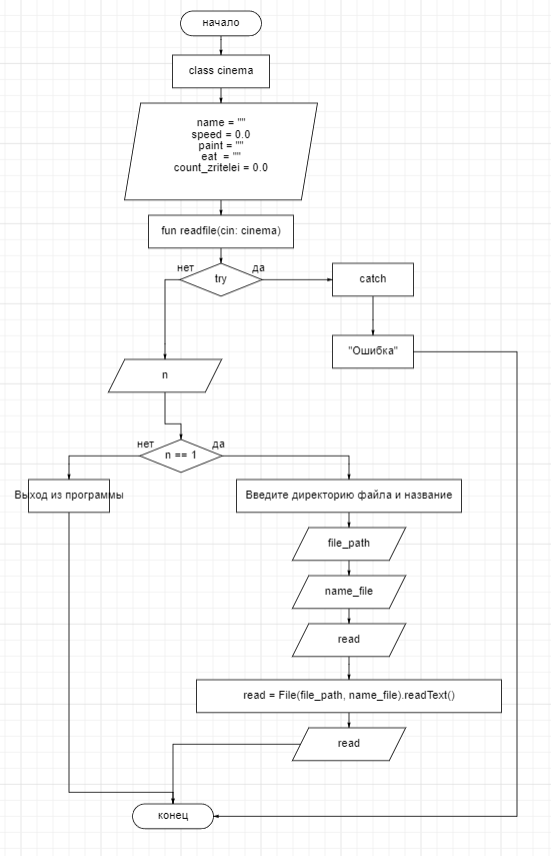
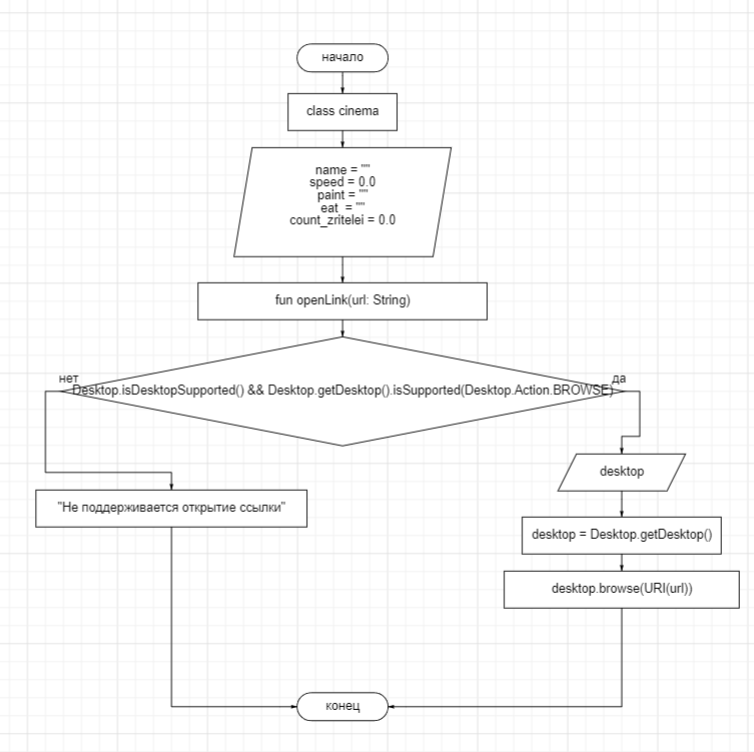
**4. **

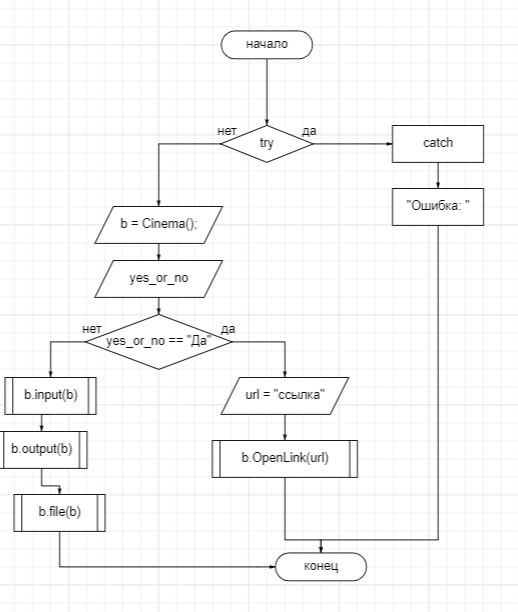
**5.**

**Main**

****

**Класс cinema**

1. ****
2. ****
3. ****
4. ****
5. ****
6. ****

**Main  
7. **

**Листинг программы (если есть)**

**Класс Вариант 2**class Nacekomie  
{  
 var name = ""  
 var speed = 0.0  
 var paint = ""  
 var eat = ""  
 var location = ""  
  
 fun input(nas:Nacekomie)  
 {  
 try  
 {  
 *println*("Введите имя насекомого")  
 nas.name = *readLine*()!!.toString()  
 *println*("Введите скорость насекомого")  
 nas.speed = *readLine*()!!.*toDouble*()  
 *println*("Введите окраску насекомого")  
 nas.paint = *readLine*()!!.toString()  
 *println*("Введите еду")  
 nas.eat = *readLine*()!!.toString()  
 *println*("Введите место обитания насекомого")  
 nas.location = *readLine*()!!.toString()  
 }  
 catch (ex:Exception)  
 {  
 *println*("Ошибка: $ex")  
 }  
  
 }  
  
 fun Output()  
 {  
 *println*("Имя насекомого: $name \nСкорость насекомого: $speed\nОкраска: $paint\nЕда: $eat\nМесто обитания: $location ")  
 }  
  
 fun Vibor()  
 {  
 *println*("Выберите:\n1 - Ввести самим:\n2 - Выбрать готовое")  
 var n = *readLine*()!!.*toInt*()  
  
 if (n == 1)  
 {  
 input(this)  
 }  
 else if (n == 2)  
 {  
 *println*("Выберите:\n1 - Кузнечик\n2 - Мошки\n3 - Стрекоза")  
 var c = *readLine*()!!.*toInt*()  
 try  
 {  
 if (c == 1)  
 {  
 name = "Кузнечик"  
 speed = 20.0  
 paint = "Зеленый"  
 eat = "Личинки, мошки"  
 location = "Поля, лес"  
 }  
 else if (c == 2)  
 {  
 name = "Мошки"  
 speed = 15.0  
 paint = "Черные"  
 eat = "Мелкие насекомые, кровь"  
 location = "Болота, леса"  
 }  
 else if (c == 3)  
 {  
 name = "Стрекоза"  
 speed = 15.0  
 paint = "Зеленая, светый-зеленый"  
 eat = "Мелкие насекомые"  
 location = "Болота, леса, поля"  
 }  
 }  
 catch (ex:Exception)  
 {  
 *println*("Ошибка")  
 }  
 }  
 else  
 {  
 *println*("Нужно выбрать либо 1 или 2")  
 System.exit(0)  
 }  
 }  
 fun File(nacekomie: Nacekomie)  
 {  
 *println*("Хотите записать в файл:\n1 - Да\n2 - Нет")  
 var l = *readLine*()!!.*toInt*()  
  
 if (l == 1)  
 {  
 var filename = *readLine*().*toString*()  
 val writer = File("${nacekomie.name}.txt").*bufferedWriter*()  
 writer.write("Имя насекомого: ${nacekomie.name}")  
 writer.newLine()  
 writer.write("Скорость насекомого: ${nacekomie.speed}")  
 writer.newLine()  
 writer.write("Окраска: ${nacekomie.paint}")  
 writer.newLine()  
 writer.write("Еда: ${nacekomie.eat}")  
 writer.newLine()  
 writer.write("Место обитания: ${nacekomie.location}")  
 writer.close()  
 *println*("Файл записан.")  
 }  
 else if (l == 2)  
 {  
 System.exit(0)  
 }  
 else  
 {  
 System.exit(0)  
 }  
  
 }  
 fun Readfile(nacekomie: Nacekomie)  
 {  
 *println*("Хотите прочитать информацию с файла: 1 - Да, 2 - Нет")  
 var n = *readLine*()!!.*toInt*()  
 if (n == 1)  
 {  
 *println*("Введите директорию файла и название: ")  
 var file\_path = *readLine*()!!.toString()  
 var name\_file = *readLine*()!!.toString()  
  
 var read = File(file\_path,name\_file).*readText*()  
 *println*(read)  
 }  
 else  
 {  
 System.exit(0)  
 }  
 }  
}

**Main**

fun main(){  
 try {  
 var b: nacekomie = nacekomie()  
 b.vibor()  
 b.output()  
 b.file(b)  
 b.readfile(b)  
 }  
 catch (ex:Exception)  
 {  
 *println*("Ошибка: ${ex.message}")  
 }  
  
}

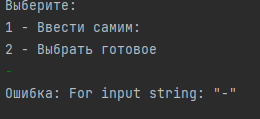
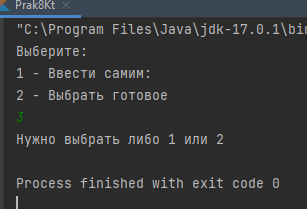
**Класс Вариант 12**class Cinema {  
 var name\_cinema = ""  
 var sianse = ""  
 var price = 0.0  
 var zal = ""  
 var count\_zritelei = 0.0  
  
 fun Input(cin: Cinema) {  
 try {  
 *println*("Введите название кинотеатра")  
 cin.name\_cinema = *readLine*()!!.toString()  
 *println*("Введите сеанс в кинотеатре")  
 cin.sianse = *readLine*()!!.toString()  
 *println*("Введите цену билета")  
 cin.price = *readLine*()!!.*toDouble*()  
 *println*("Введите зал")  
 cin.zal = *readLine*()!!.toString()  
 *println*("Введите количество зрителей")  
 cin.count\_zritelei = *readLine*()!!.*toDouble*()  
 } catch (ex: Exception) {  
 *println*("Ошибка")  
 }  
 }  
  
 fun Output() {  
 *println*("Кинотеатр: $name\_cinema\nСеанс: $sianse\nЦена: $price\nЗал: $zal\nКоличество зрителей: $count\_zritelei")  
 }  
  
 fun File(cin: Cinema) {  
 *println*("Хотите записать в файл:\n1 - Да\n2 - Открыть файл\n3 - Закрыть программу")  
 var l = *readLine*()!!.*toInt*()  
  
 if (l == 1) {  
 var filename = *readLine*().*toString*()  
 val writer = File("${cin.name\_cinema}.txt").*bufferedWriter*()  
 writer.write("Название кинотеатра: ${cin.name\_cinema}")  
 writer.newLine()  
 writer.write("Сеанс: ${cin.sianse}")  
 writer.newLine()  
 writer.write("Цена билета: ${cin.price}")  
 writer.newLine()  
 writer.write("Зал: ${cin.zal}")  
 writer.newLine()  
 writer.write("Количество зрителей: ${cin.count\_zritelei}")  
 *println*("Файл записан.")  
 writer.close()  
 }  
 if (l == 2) {  
 Readfile(this)  
 } else {  
 System.exit(0)  
 }  
  
 }  
  
 fun Readfile(cin: Cinema)  
 {  
 try {  
 *println*("Хотите прочитать информацию с файла: 1 - Да, 2 - Нет")  
 var n = *readLine*()!!.*toInt*()  
 if (n == 1)  
 {  
 *println*("Введите директорию файла и название: ")  
 var file\_path = *readLine*()!!.toString()  
 var name\_file = *readLine*()!!.toString()  
  
 var read = File(file\_path, name\_file).*readText*()  
 *println*(read)  
 }  
 else  
 {  
 System.exit(0)  
 }  
 }  
 catch (ex:Exception)  
 {  
 *println*("Ошибка: ${ex.message}")  
 }  
  
  
  
  
 }  
  
 fun OpenLink(url: String)  
 {  
 if (Desktop.isDesktopSupported() && Desktop.getDesktop().isSupported(Desktop.Action.*BROWSE*))  
 {  
 val desktop = Desktop.getDesktop()  
 desktop.browse(URI(url))  
 }  
 else  
 {  
 *println*("Не поддерживается открытие ссылки")  
 }  
 }  
}

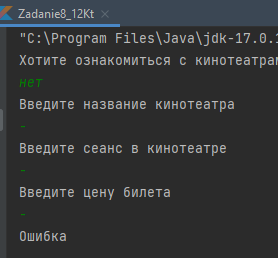
**Main**

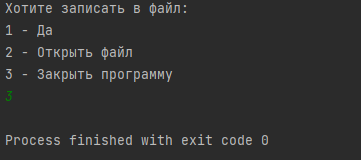
fun main(){  
 try {  
 var b = Cinema();  
 *println*("Хотите ознакомиться с кинотеатрами?: ")  
 var yes\_or\_no = *readLine*()!!.toString()  
  
 if (yes\_or\_no == "Да")  
 {  
 val url = "https://www.tripadvisor.ru/Attractions-g294459-Activities-c56-t97-Russia.html"  
 b.OpenLink(url)  
 }  
 else  
 {  
 b.Input(b)  
 b.Output()  
 b.File(b)  
 }  
 }  
 catch (ex:Exception)  
 {  
 *println*("Ошибка: ${ex.message}")  
 }  
}

**Тестовые ситуации**

**Вариант 2**

**  
**

**Вариант 12  
**

****

**Вывод**

Задание 1: Прочитать букву с экрана и вывести на печать, предыдущую перед ней букву в алфавитном порядке.

Листинг программы:

using System;

namespace Практика\_задание1\_1

{

class Program

{

static void Main(string[] args)

{

string letter = Console.ReadLine();

string alphav = "АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ";

string alphav\_1 = alphav.ToLower();

int point = alphav.IndexOf(letter);

int point\_1 = alphav\_1.IndexOf(letter);

if (alphav.Contains(letter))

{

Console.WriteLine(alphav[point + 1]);

}

if (alphav\_1.Contains(letter))

{

Console.WriteLine(alphav\_1[point\_1 + 1]);

}

}

}

}