**Лабораторная работа №9**

**Основы программирования на Java: операторы выбора**

**Цель работы:** Освоить основные способы создания Java-программ либо с помощью обычного редактора, либо с помощью среды разработки. Приобрести навыки работы с операторами выбора в Java.

**Теоретические сведения**

**Java — Оператор switch..case**

**Оператор switch** — проверяет переменную на равенство в отношении списка значений. Каждое значение называется case, и переменная переключаясь проверяется для каждого case.

**Синтаксис**

Синтаксис для расширенного цикла switch в Java следующий:

switch(выражение){

case значение :

//Операторы

break; //необязательно

case значение :

//Операторы

break; //необязательно

//Вы можете иметь любое количество операторов case.

default : //необязательно

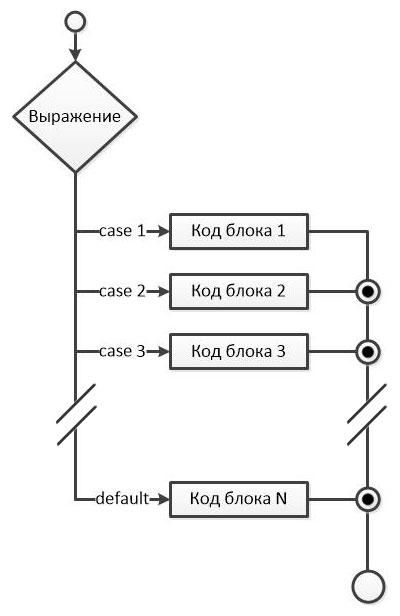
//Операторы

}

Следующие правила применяются для оператора switch:

* Переменные, которые используются в операторе switch, могут быть только целые числа, конвертированные в целые числа (byte, short, char), строки и перечисления.
* Вы можете иметь любое количество операторов case в рамках одного switch. За каждым case следует сравниваемое значение, а затем идёт двоеточие.
* Значение case должен быть того же типа данных как и переменная в switch, и она должна быть константой или литерал.
* Когда переменная switch на равна оператору case, операторы следующие за case будут выполняться до тех пор, пока не будет достигнут оператор break.
* При достижении оператора break, switch завершается, и поток управления переходит к следующей строке после оператора switch.
* Не каждый case должен содержать break. Если отсутствует break, поток управления попадет на следующие case, до тех пор пока break не будет достигнут.
* В Java оператор switch может иметь дополнительный default case, который должен находиться в конце switch. Default case может быть использован для выполнения задачи, когда ни один из case является правильным. Break не требуется в default case.

**Описание процесса**



**Пример**

public class Test {

public static void main(String args[]){

//char grade = args[0].charAt(0);

char grade = 'C';

switch(grade)

{

case 'A' :

System.out.println("Отлично!");

break;

case 'B' :

case 'C' :

System.out.println("Отлично выполнено");

break;

case 'D' :

System.out.println("Вы прошли");

case 'F' :

System.out.println("Лучше попробуйте снова");

break;

default :

System.out.println("Неверная оценка");

}

System.out.println("Ваша оценка " + grade);

}

}

Скомпилируйте и запустите java-программу, используя различные аргументы в командной строке. Будет получен следующий результат:

Отлично выполнено

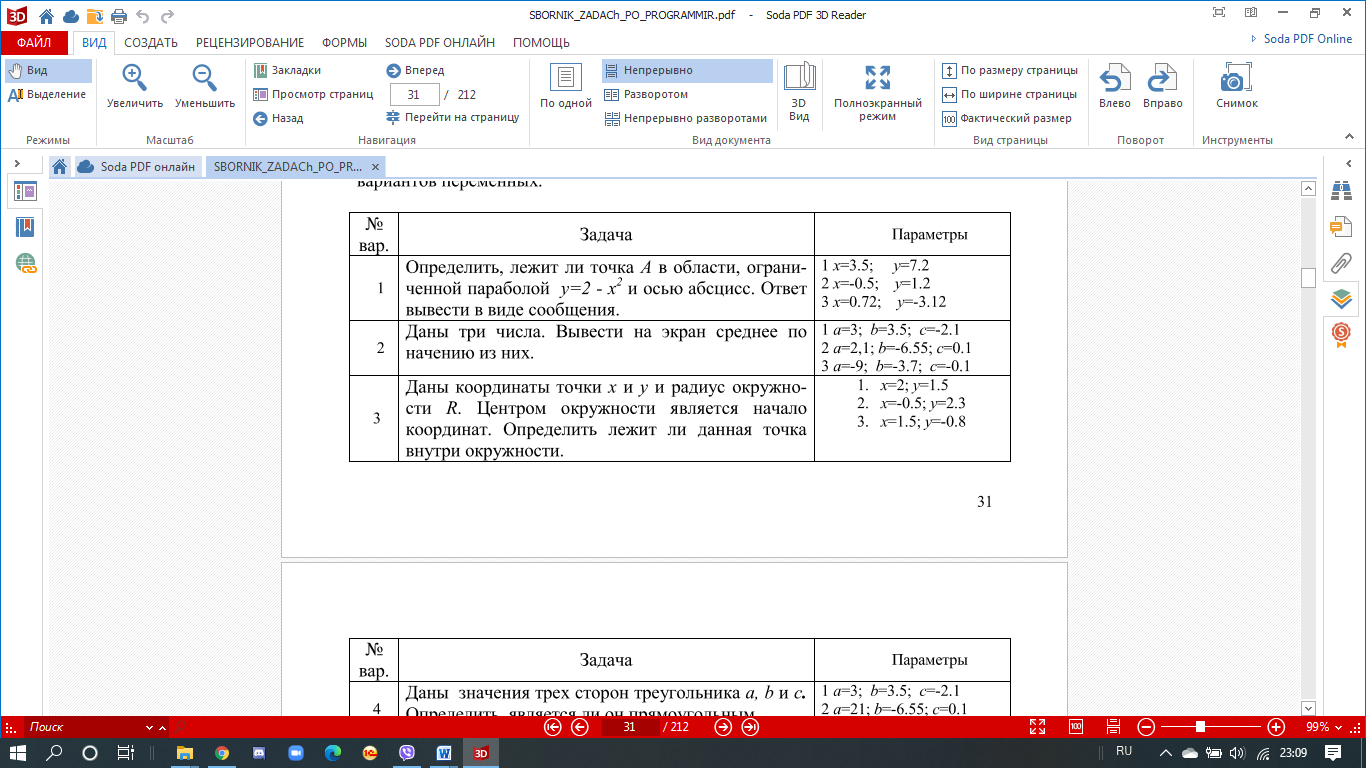
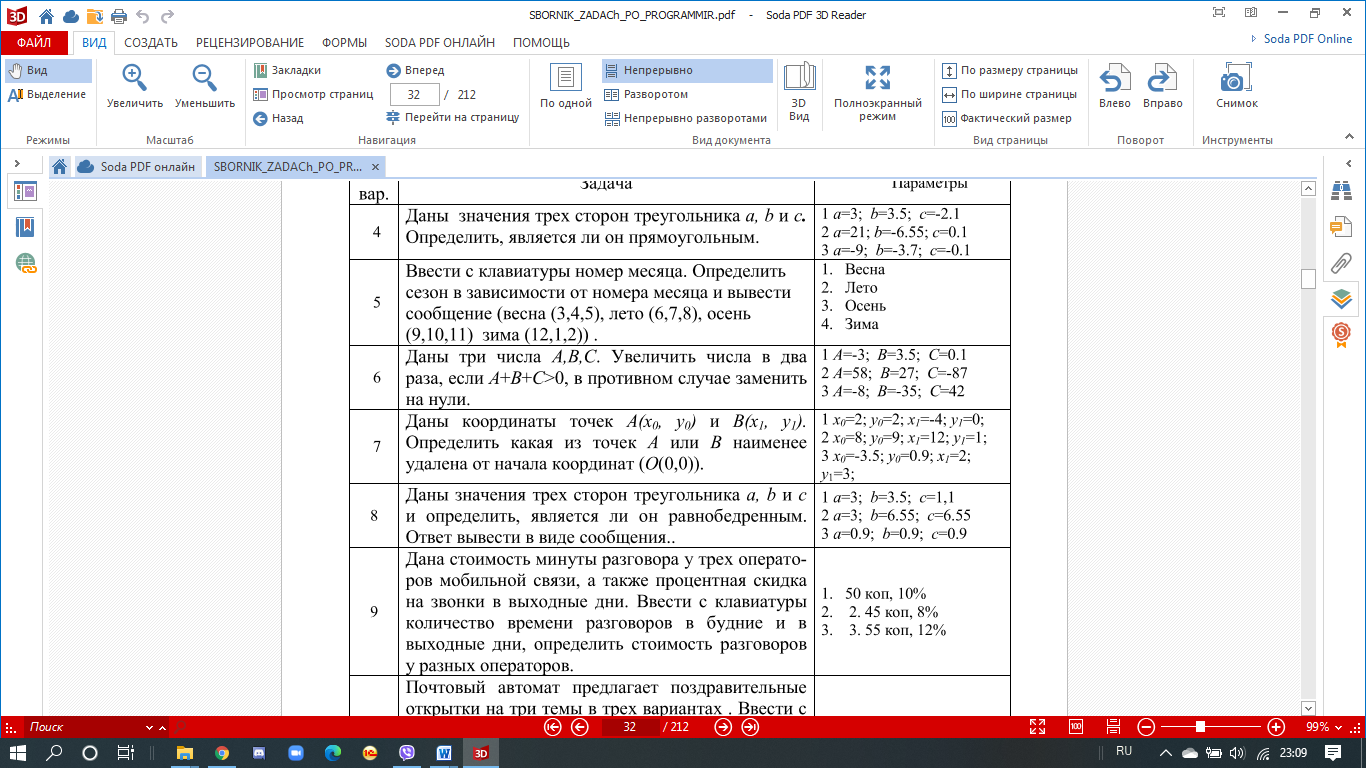
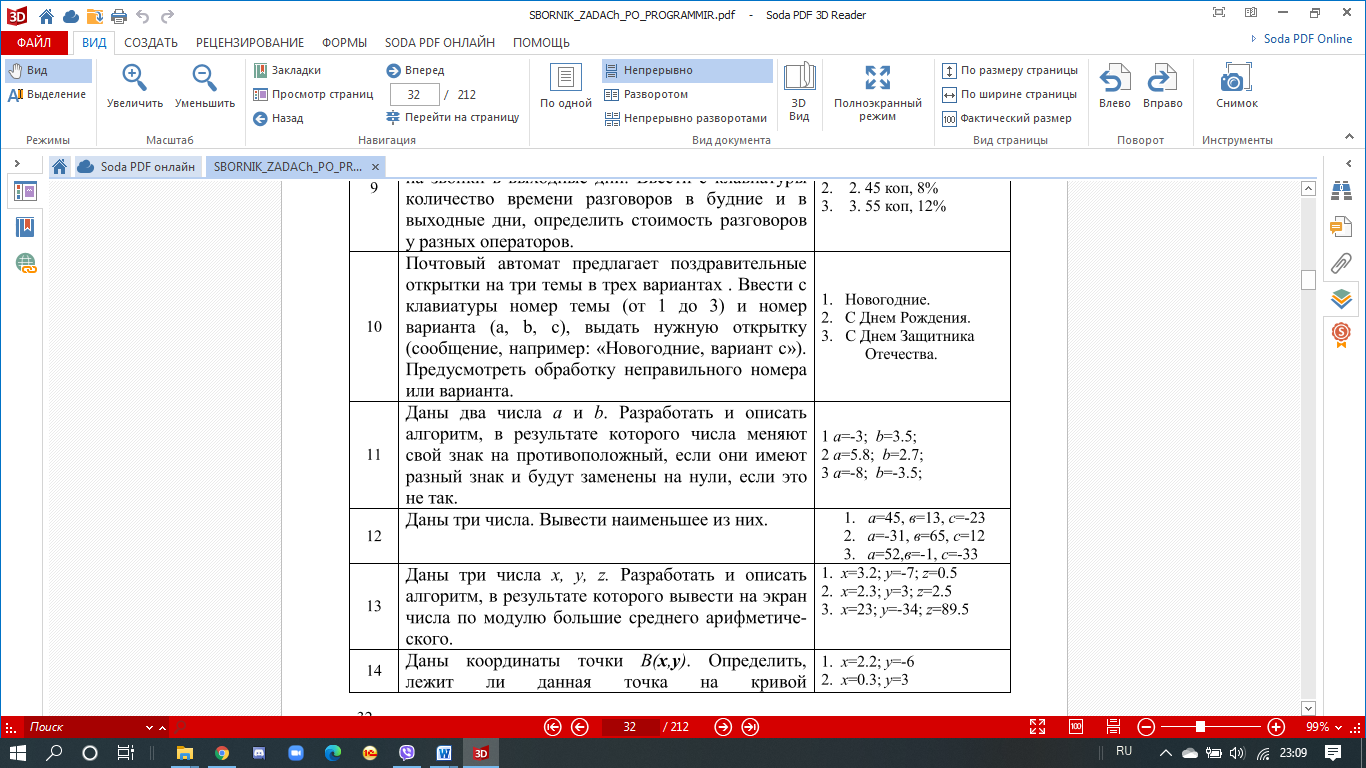
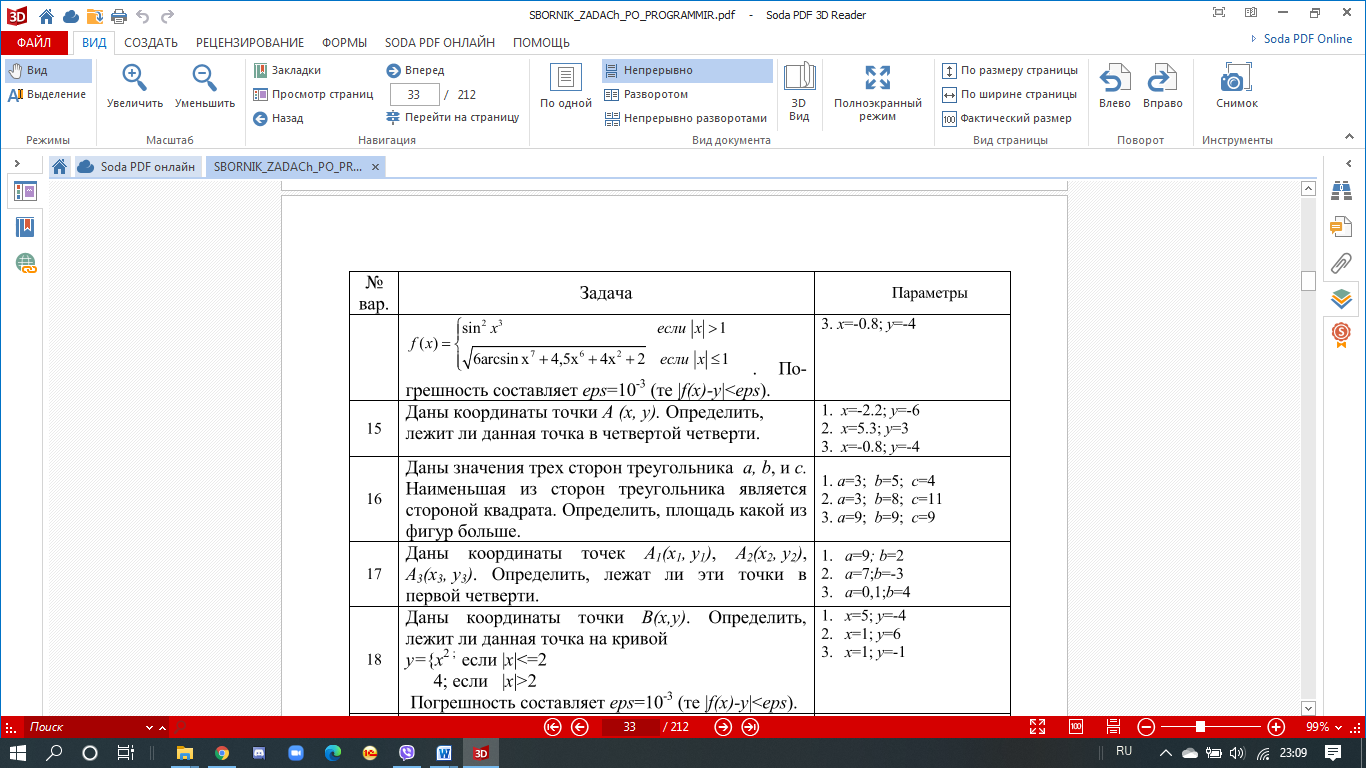
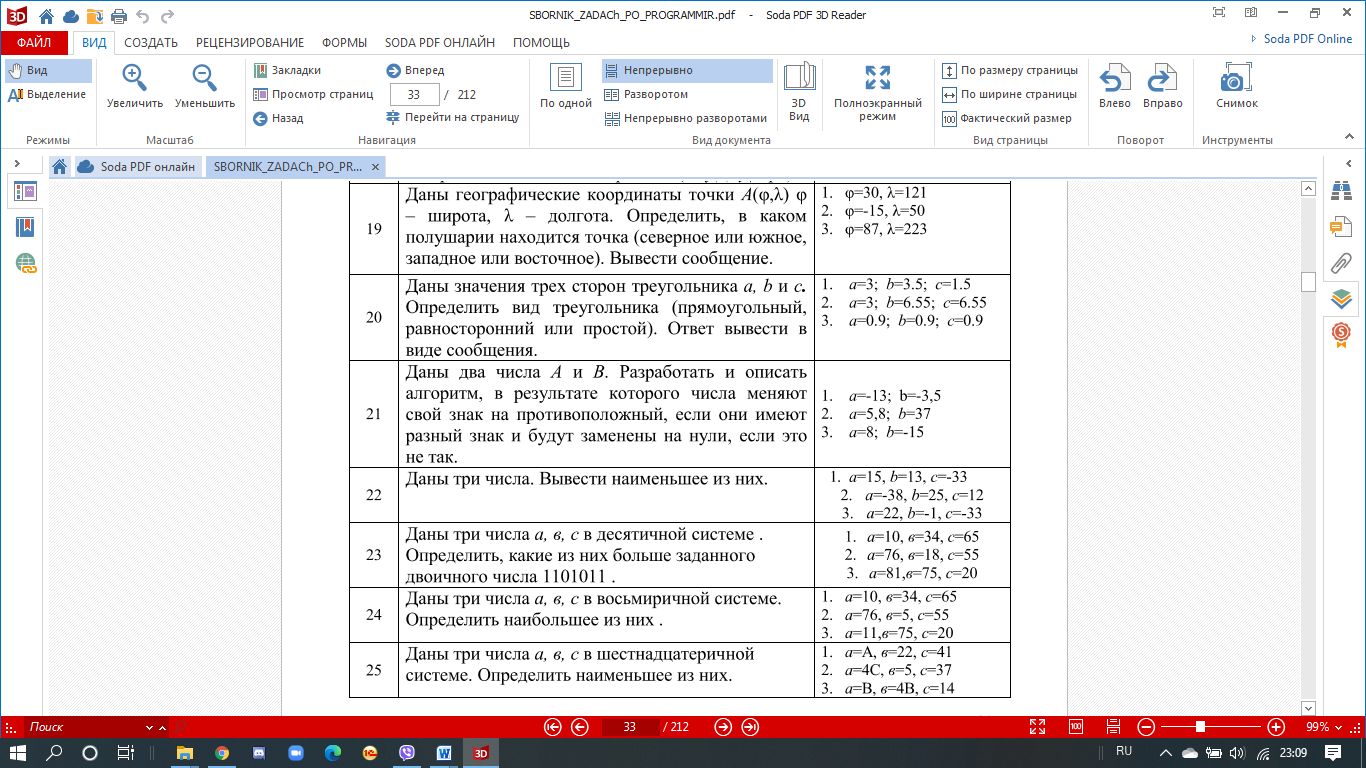
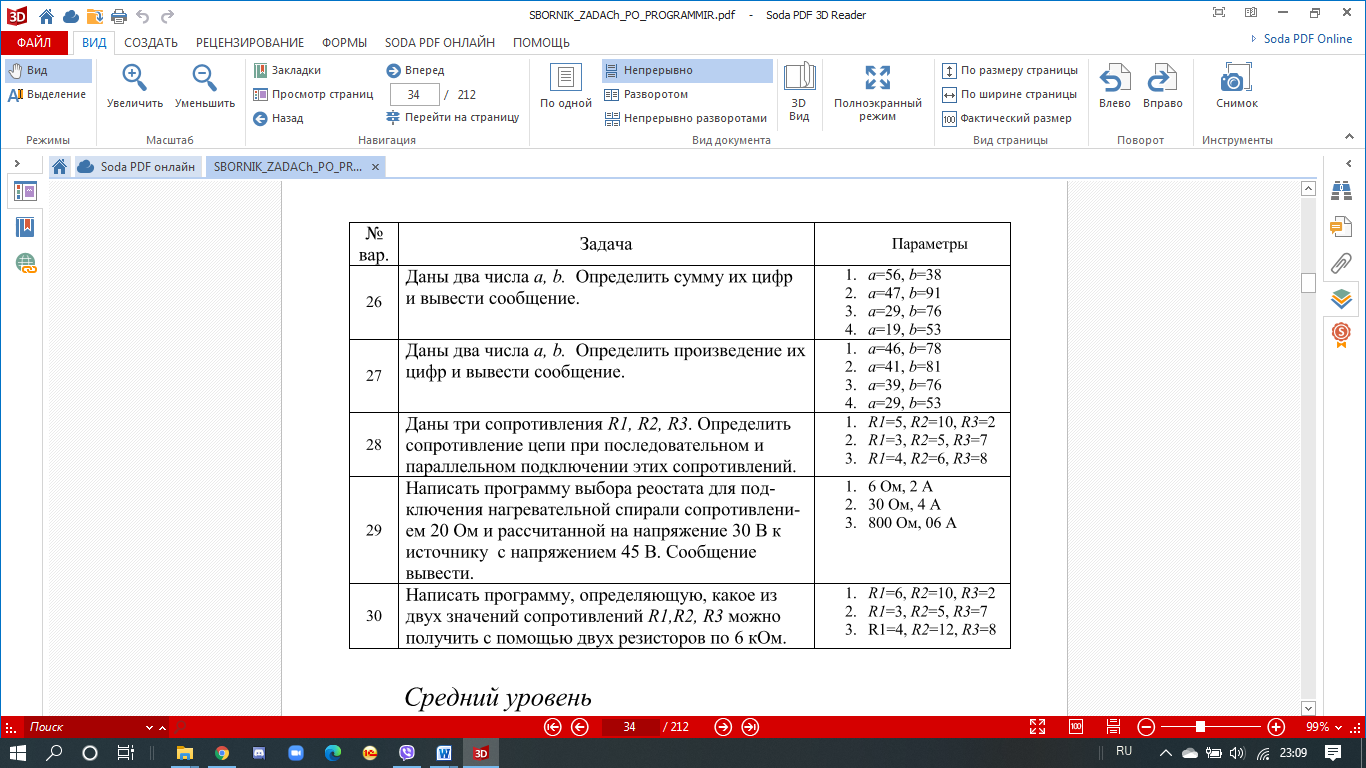
Ваша оценка C

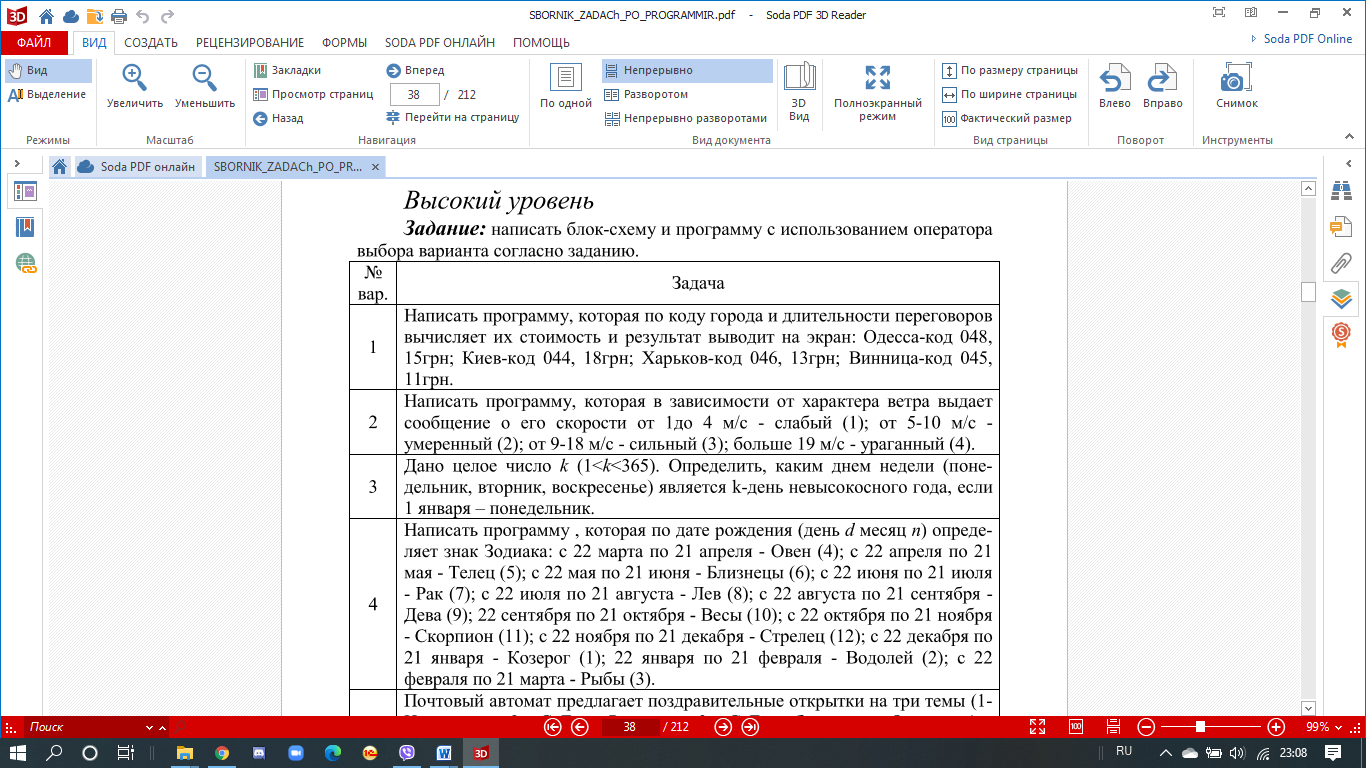
**Задание**

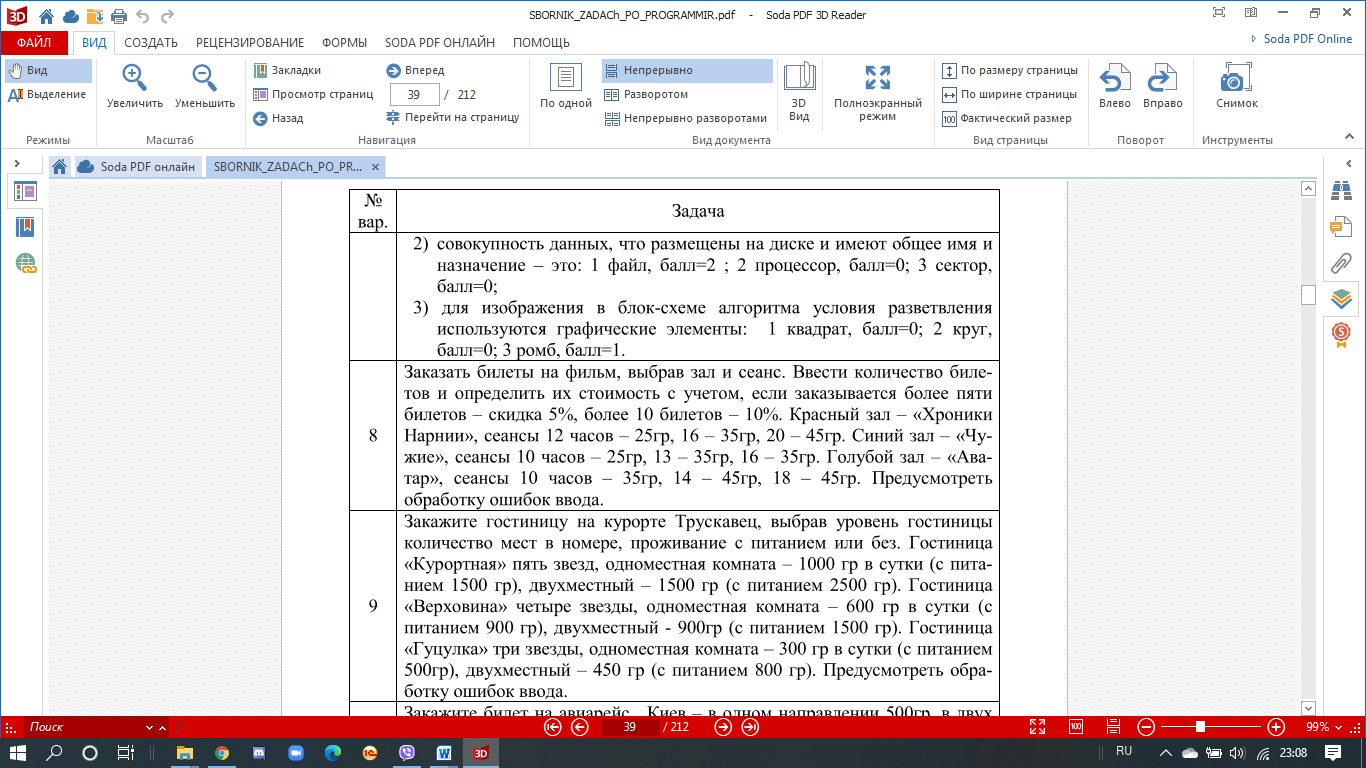
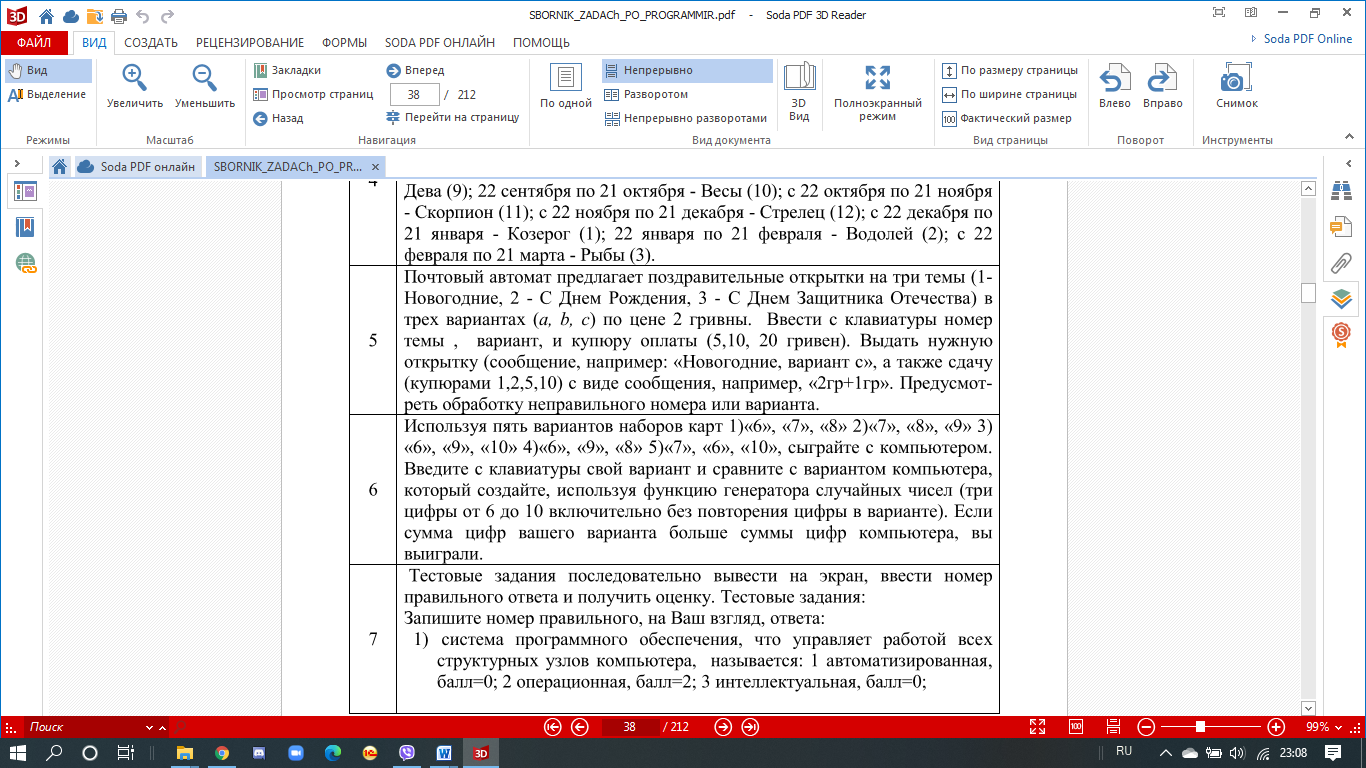
Изучить теоретический материал и выполнить задачи по вариантам.

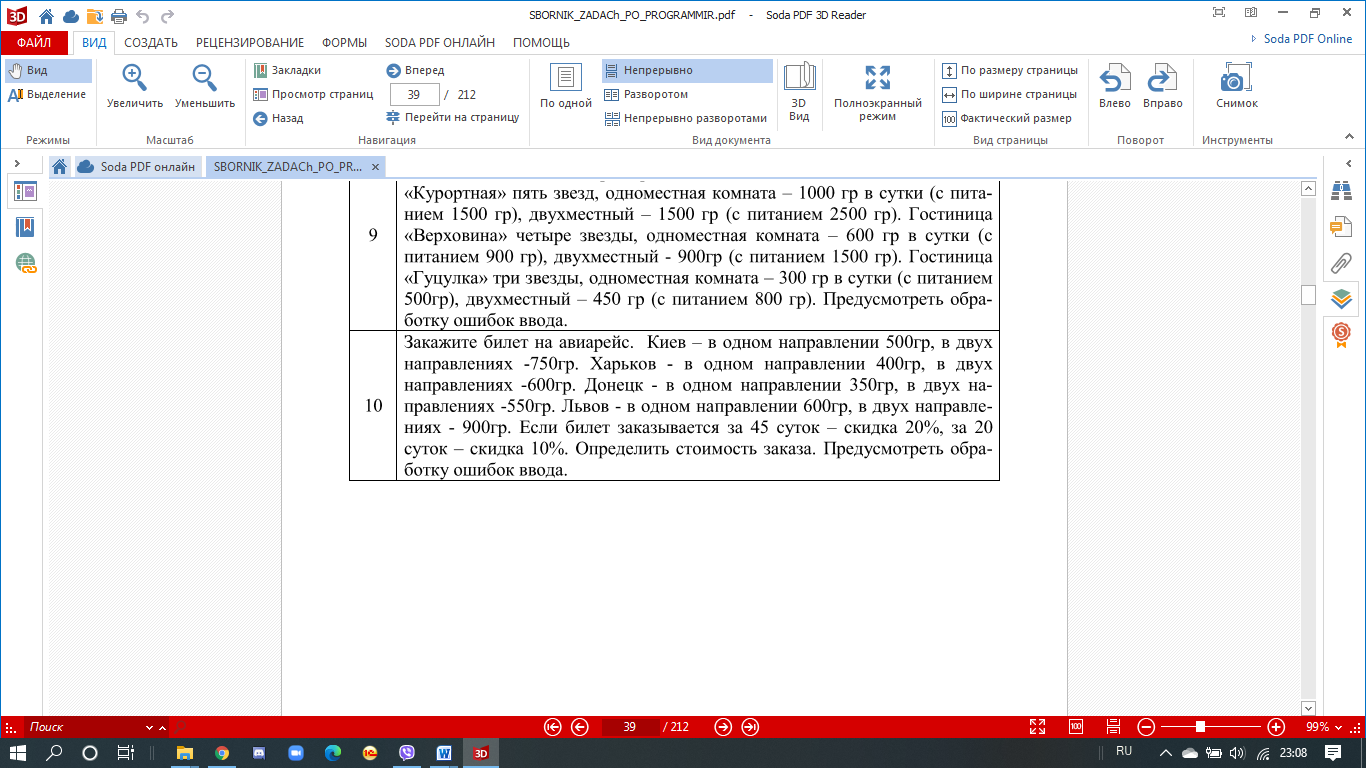
Составить блок-схему алгоритма и проект программы решения задачи, исходные данные ввести с клавиатуры.

**Задача 1**

       
**Задание 2**







**Форма отчета**

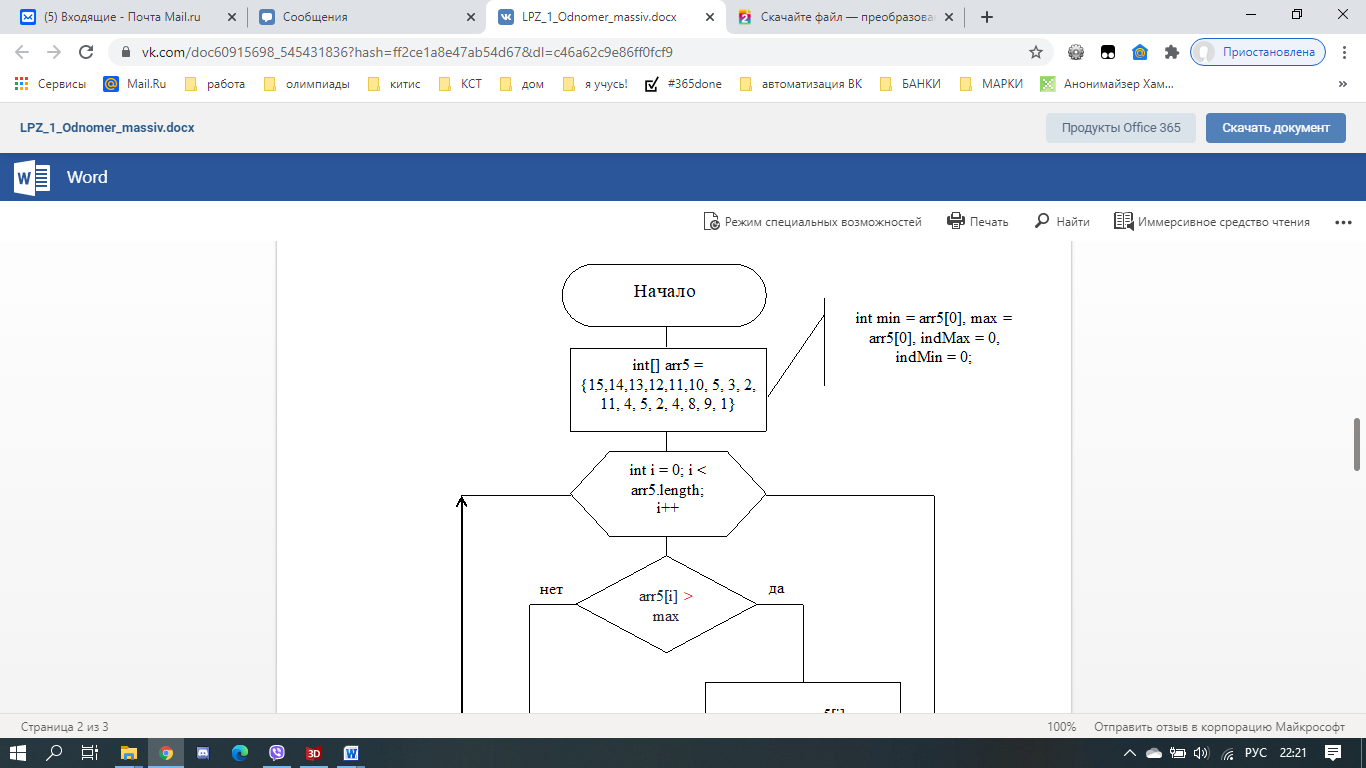
1. Тема, цель, индивидуальное задание

2. Таблица идентификаторов

|  |  |  |
| --- | --- | --- |
| Наименование переменной | Тип данных | Назначение |
|  |  |  |
|  |  |  |
|  |  |  |

3. Блок-схема алгоритма (см. Приложение)

Пример:



4. Код программы

5. Скриншот результата

**Контрольные вопросы для защиты**

1. Оператор if?
2. Оператор switch..case. Примеры.
3. Вложенные операторы.

**Приложение**

**Условные графические обозначения в схемах алгоритмов и программ, отображающие основные операции процесса обработки данных и программирования по ГОСТ 2.708 - 81**

Размер, а следует выбирать из ряда 10, 15, 20 мм. Допускается увеличение размера а на число, кратное 5. Размер b принима­ют равным 1,5а.

|  |  |
| --- | --- |
|  | Процесс. Выполнение операций (группы операций), в результате которых изменяется значение (форма представления, расположение) данных  Решение. Выбор направления алгоритма (програм­мы) в зависимости от некоторых переменных условий |
|  | Решение. Выбор направления алгоритма (програм­мы) в зависимости от некоторых переменных условий |
|  | Модификация. Выполнение операций, меняющих команды (группы команд), изменяющих программу |
|  | Предопределенный процесс. Использование ранее созданных и описанных отдельно алгоритмов (программ) |
|  | Ввод-вывод. Преобразование данных в форму, пригодную для обработки (ввод) или отображения результатов обработки (вывод) |
|  | Соединитель. Указание связи между прерванны­ми линиями потока (связывющие символы) |
|  | Пуск-останов. Начало, конец, прерывание про­цесса обработки данных или выполнения программы |
|  | Межстрочный соединитель |