**Лабораторная работа №7**

**Основы программирования на Java: операторы принятия решений**

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

**Теоретические сведения**

Структура принятия решения состоит из одного или нескольких условий, которые проверяются в рамках программы вместе с оператором или операторами, которые должны быть выполнены, если условие определяется как истинно, и, возможно, другие операторы, которые будут выполняться, если условие определяется как ложно.

Ниже приводится общий вид типичной структуры принятия решений, используемая в большинстве языков программирования:



Язык программирования Java предлагает следующие виды операторов принятия решений:

|  |  |
| --- | --- |
| **Оператор** | **Описание** |
| [if](http://proglang.su/java/operator-if) | Оператор if состоит из логического выражения, после которого следует один или более операторов. |
| [if..else](http://proglang.su/java/operator-if-else) | Оператор if может сопровождаться дополнительным оператором else, который выполняется при ложном логическом выражении. |
| [условный (? :)](http://proglang.su/java/conditional-operator) | Вы можете использовать для замены операторов if...else. |
| [вложенный if](http://proglang.su/java/embedded-operator-if) | Вы можете использовать один оператор if или оператор if...else внутри другого или других операторов if или if...else. |

**Java — Оператор if**

**Оператор if** — состоит из логического выражения, после которого следует один или более операторов.

**Синтаксис**

Синтаксис оператора if в Java следующий:

If (Логическое выражение)

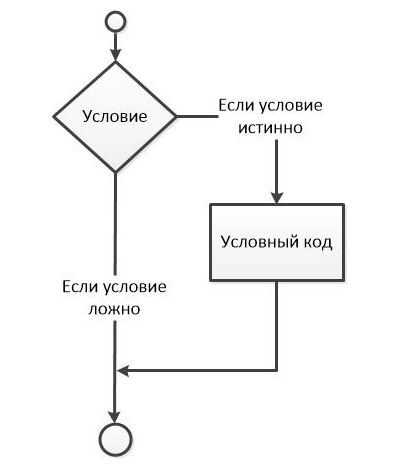
{

//Операторы выполняются, если истинно

}

Если логическое выражение истинно, то блок кода внутри оператора if будет выполнен. Если нет, то будет выполняться следующий блок кода, идущий после оператора if (после закрывающей фигурной скобки).

**Описание процесса**



**Пример**

public class Test {

public static void main(String args[]){

int x = 10;

if( x < 20 ){

System.out.print("Это оператор if");

}

}

}

Будет получен следующий результат:

Это оператор if

**Java — Оператор if..else**

В Java оператор if может сопровождаться дополнительным оператором else, который выполняется при ложном логическое выражение.

**Синтаксис**

Синтаксис оператора if..else в Java следующий:

If (Логическое выражение)

{

//Выполняется, если истинно

} else {

//Выполняется, если ложно

}

Если логическое выражение истинно, то блок кода if будет выполнен, в противном случае будет выполнен блок кода else.

**Описание процесса**



**Пример**

public class Test {

public static void main(String args[]){

int x = 30;

if( x < 20 ){

System.out.print("Это оператор if");

}else{

System.out.print("Это оператор else");

}

}

}

Будет получен следующий результат:

Это оператор else

**Оператор if...else if...else**

Оператор if может сопровождаться дополнительным оператором else if..else, который является очень полезным для тестирования различных условиях с использованием одного оператора if..else if.

При использовании операторов if, else if, else в Java есть несколько моментов, которые необходимо иметь в виду:

* if может иметь ноль или один else, который должен идти после любого if.
* if может иметь ноль для многих else, и он должен идти до else.
* после того как if...else будет успешно выполнен, из оставшихся else if или else ничто не будет выполнено (проверено).

**Пример**

public class Test {

public static void main(String args[]){

int x = 30;

if( x == 10 ){

System.out.print("Значение X = 10");

}else if( x == 20 ){

System.out.print("Значение X = 20");

}else if( x == 30 ){

System.out.print("Значение X = 30");

}else{

System.out.print("Это оператор else");

}

}

}

Будет получен следующий результат:

Значение X = 30

**Java — Условный оператор ? : (вопросительный знак и двоеточие)**

В предыдущем уроке мы рассмотрели условный оператор ? : (знак вопроса и двоеточие) в Java, который может быть использован для замены операторов if..else. Он имеет следующий общий вид:

Exp1 ? Exp2 : Exp3;

Где Exp1, Exp2 и Exp3 являются выражениями. Обратите внимание на использование и размещение двоеточия.

Чтобы определить значение всего выражения, сперва нужно оценить Exp1:

* Если значение Exp1 верно, то значение Exp2 будет значением всего выражения.
* Если значение Exp1 ложно, то вычисляется Exp3 и его значение становится значением всего выражения.

**Java — Вложенный оператор if**

В Java всегда допустимы вложенные операторы if-else, которые означают, что Вы можете использовать один оператор if или else внутри другого оператора if или else.

**Синтаксис**

Синтаксис вложенного оператора if..else в Java следующий:

If (Логическое выражение 1){

//Выполняется, если логическое выражение 1 истинно

If (Логическое выражение 1){

//Выполняется, если логическое выражение 2 истинно

}

}

Вы можете делать вложенным else if...else аналогично тому, как мы делали вложенным оператор if в Java.

**Пример**

public class Test {

public static void main(String args[]){

int x = 30;

int y = 10;

if( x == 30 ){

if( y == 10 ){

System.out.print("X = 30 и Y = 10");

}

}

}

}

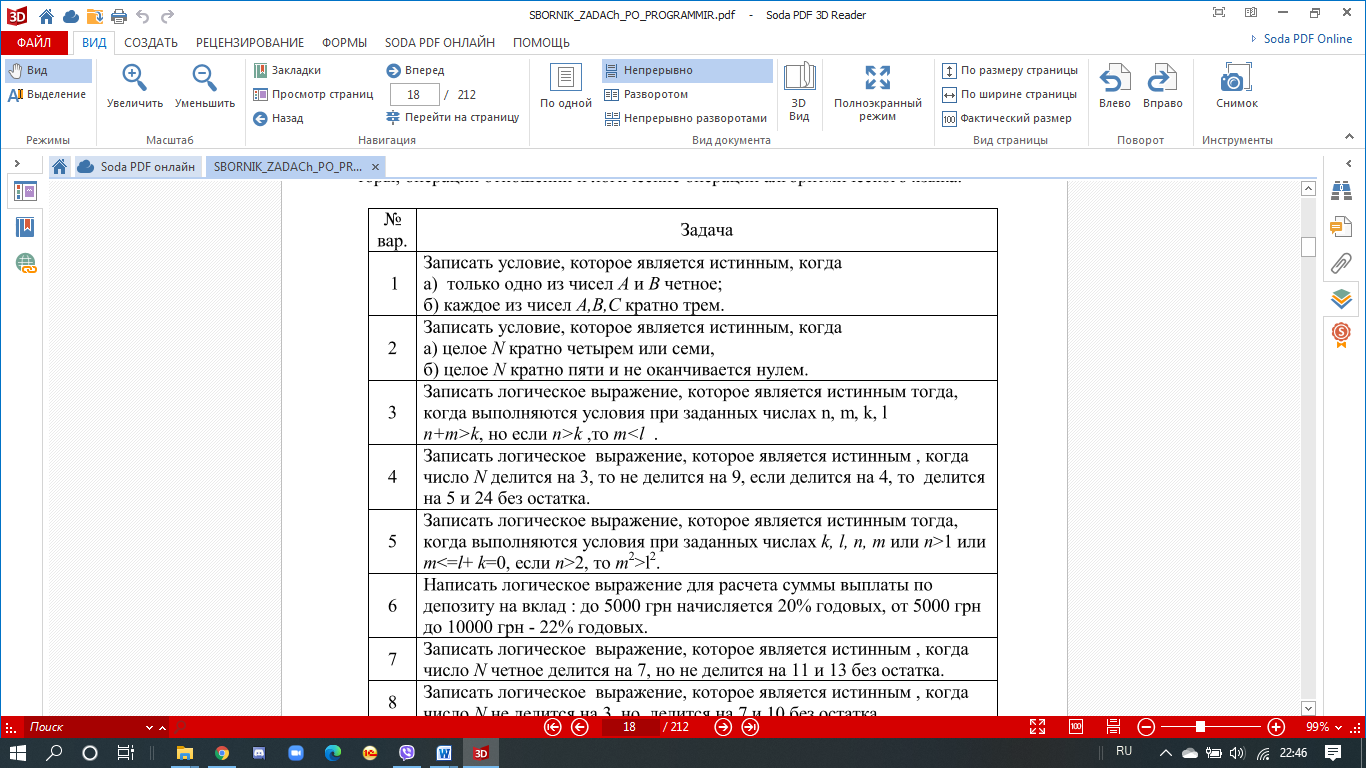
Будет получен следующий результат:

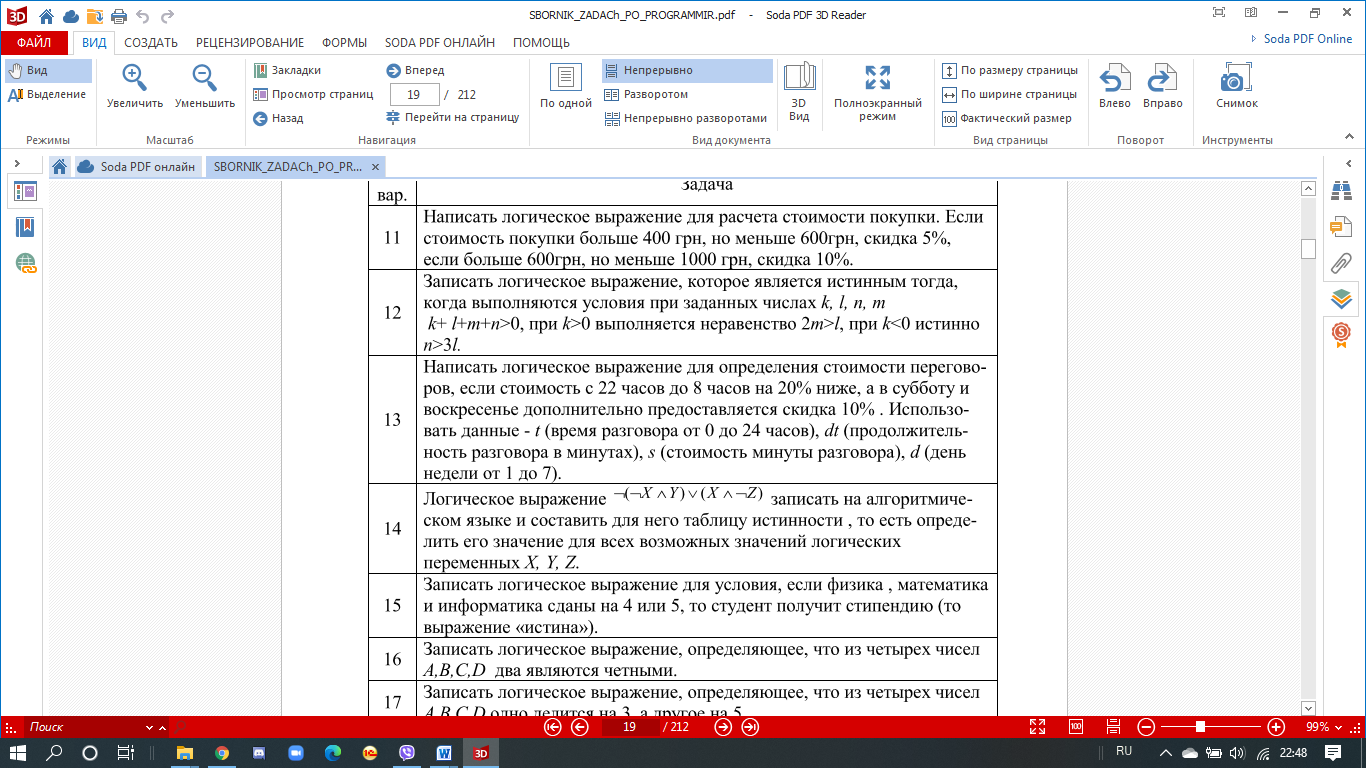
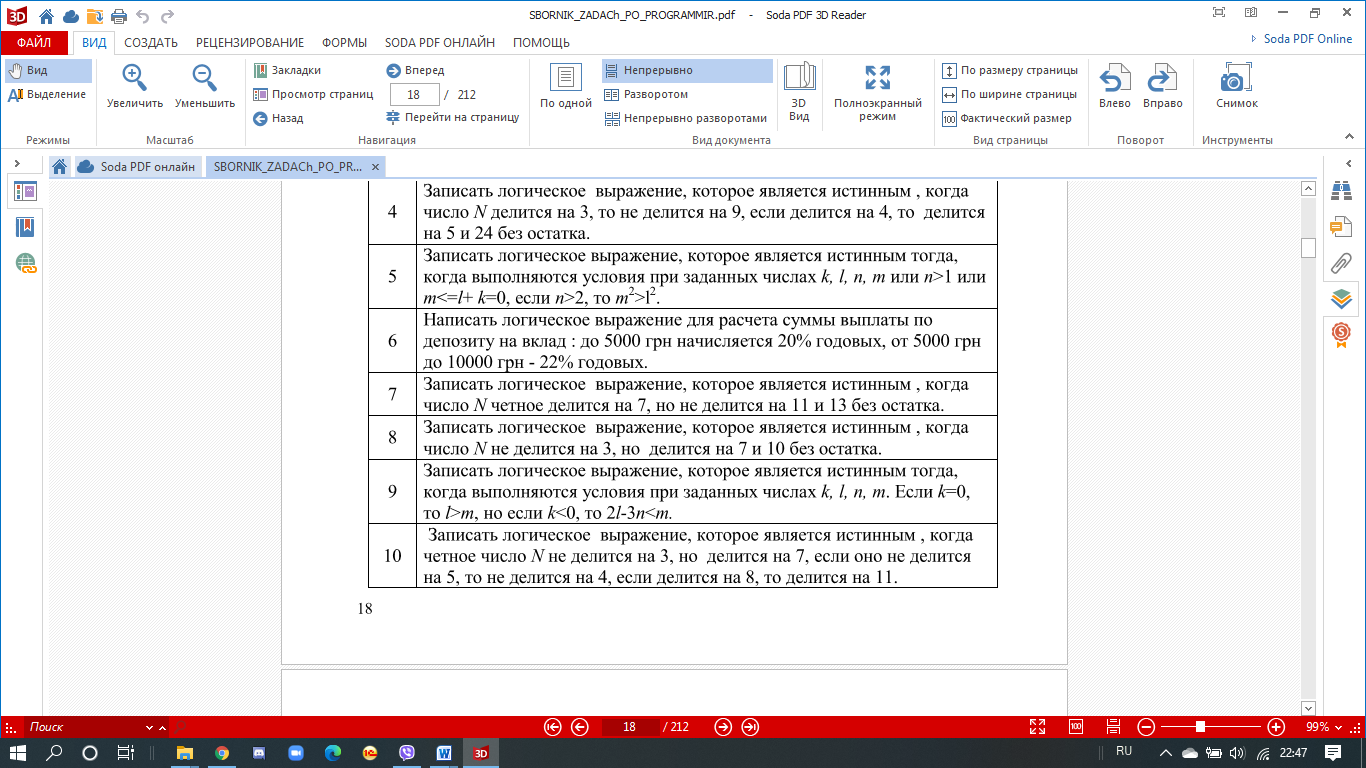
X = 30 и Y = 10

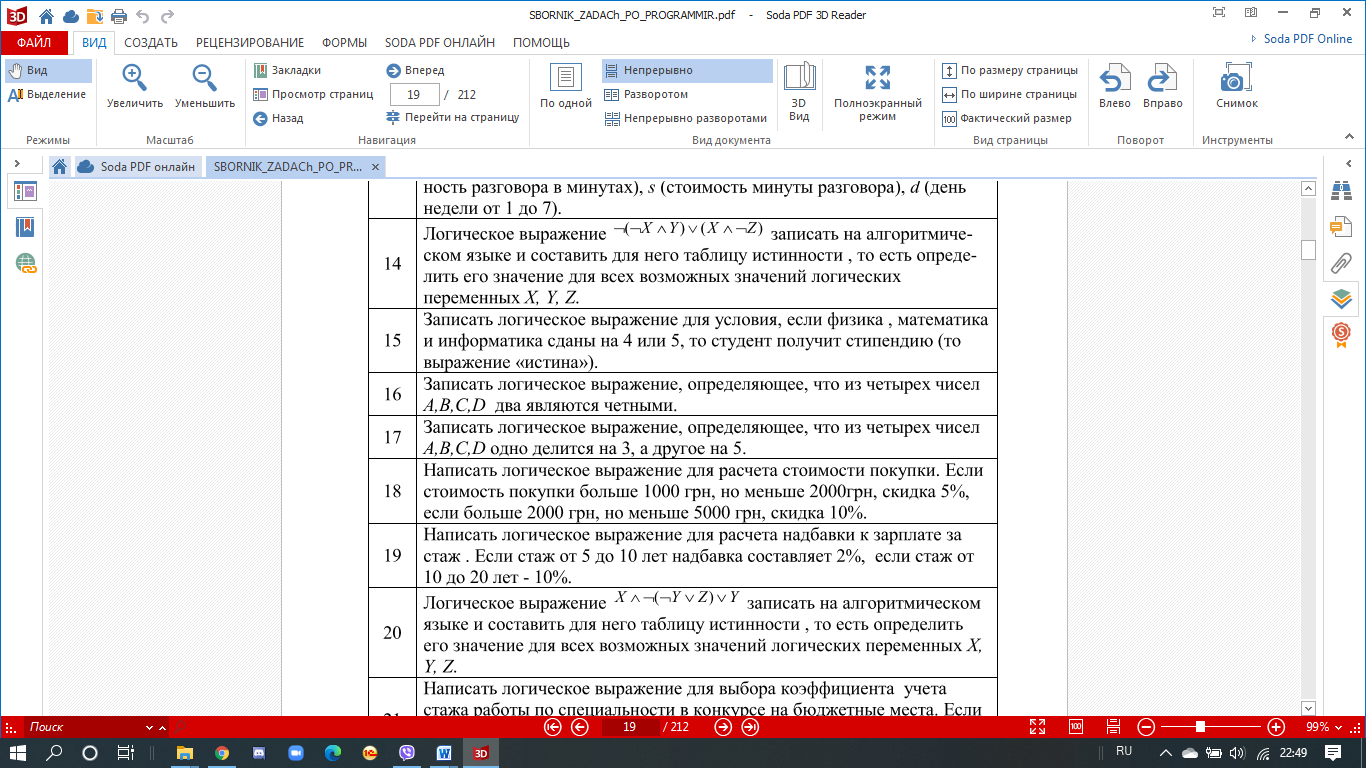
**Задание**

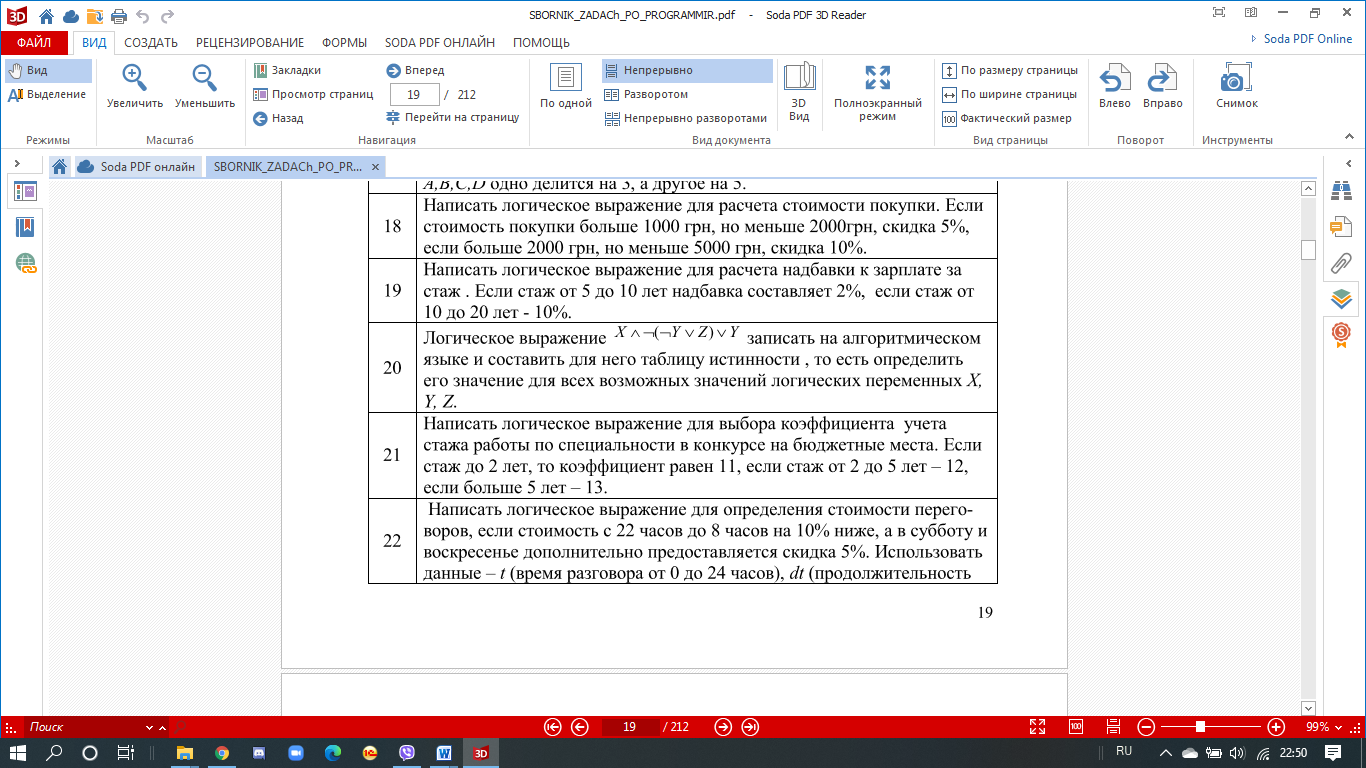
Изучить теоретический материал и выполнить задачи по вариантам.

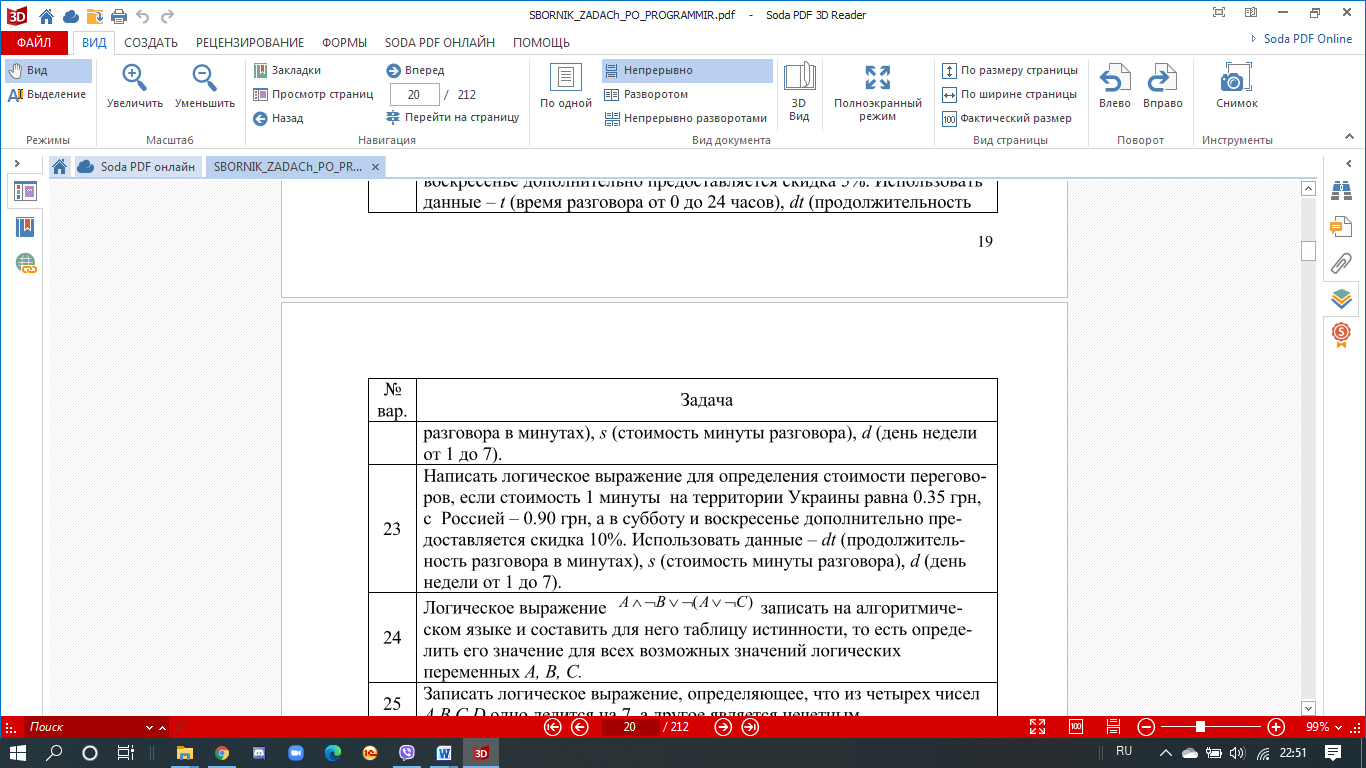
Составить блок-схему алгоритма и проект программы решения задачи, исходные данные ввести с клавиатуры.

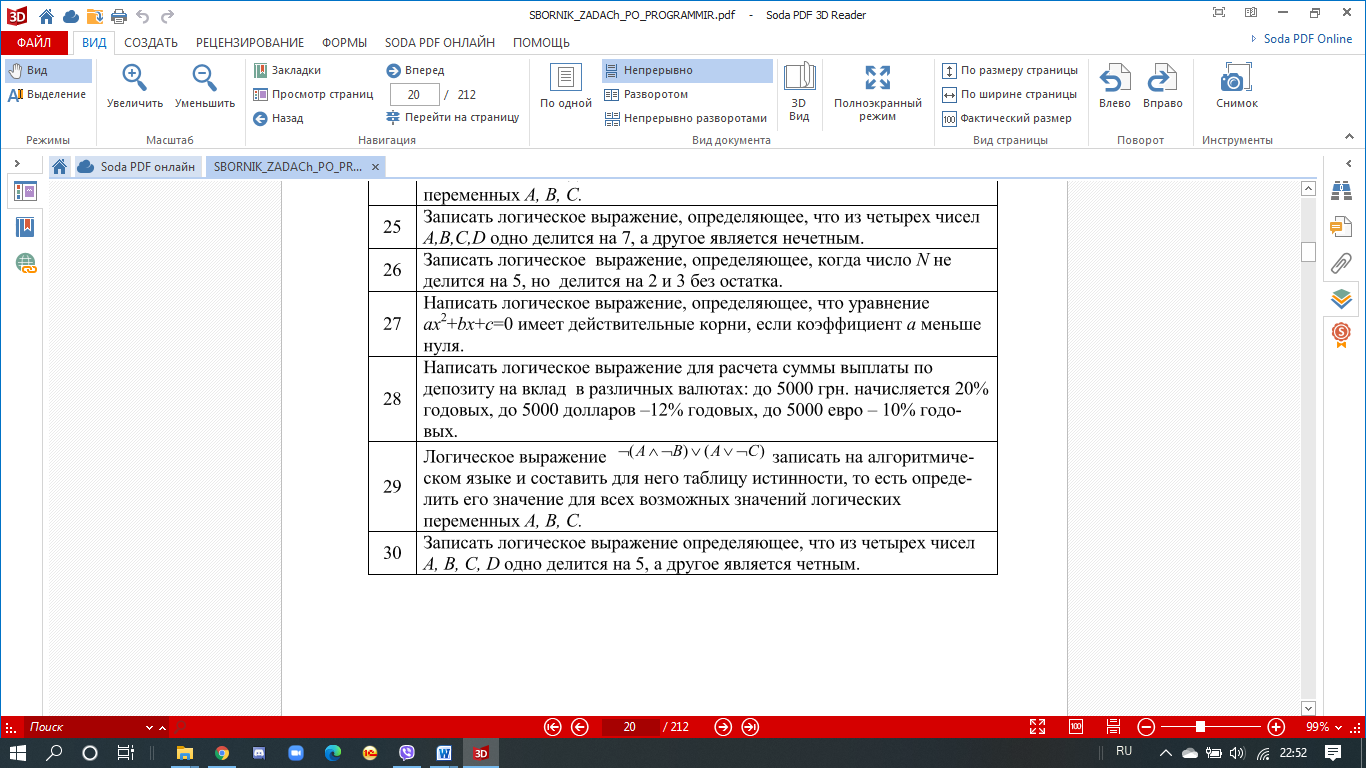












**Форма отчета**

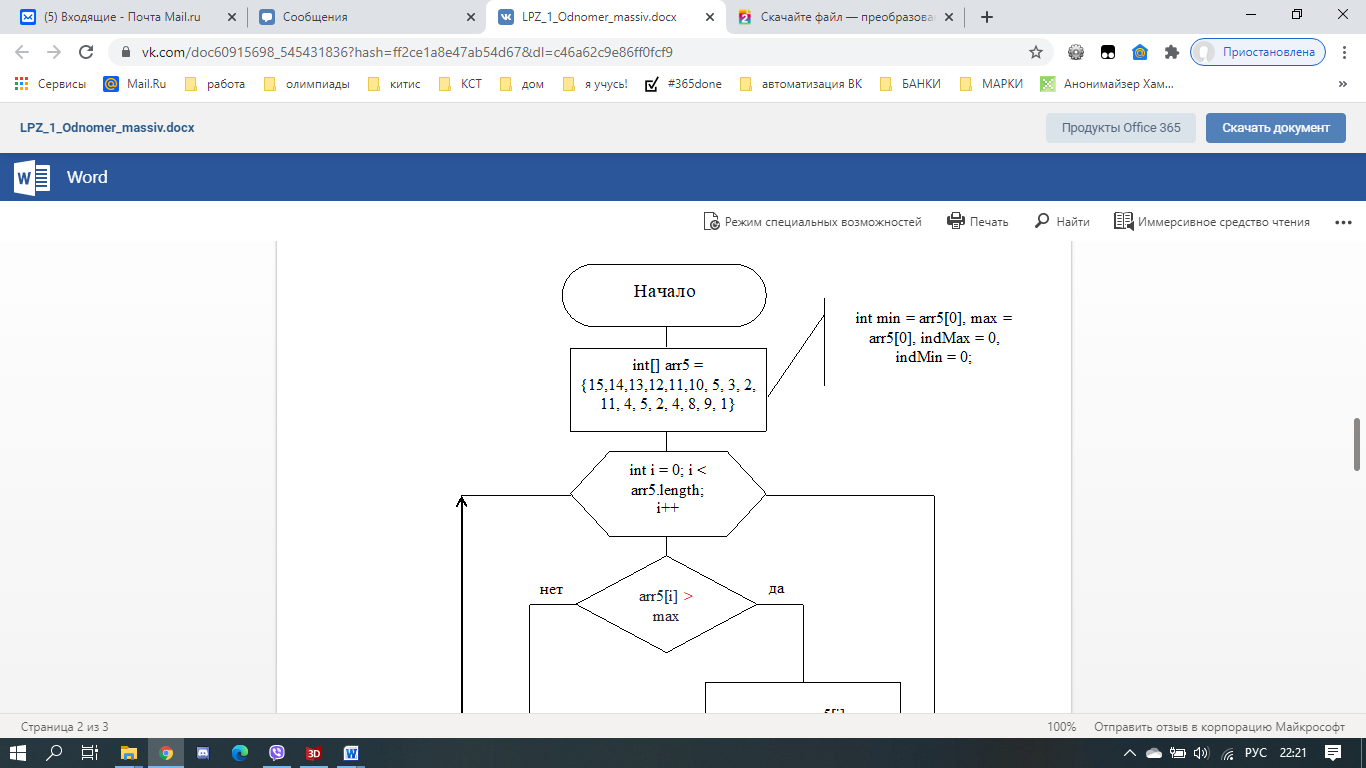
1. Тема, цель, индивидуальное задание

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. If…else в Java. Примеры.
3. Вложенные операторы IF.

**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 |