Министерство образования Республики Беларусь Белорусский Национальный Технический Университет

Факультет информационных технологий и робототехники

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

# Отчёт

по лабораторной работе №07

по дисциплине ***«Объектно-ориентированное программирование»***

тема: «Базовый синтаксис языка Java. Условные конструкции. Разветвляющиеся алгоритмы»

Название бригады: «**Best progger**»

Исполнитель: студенты группы 10701118

Воробей Иван Александрович

Преподаватель: Иванченко Виктор Викторович

2018-2019 учебный год

## ЛАБОРАТОРНАЯ РАБОТА №07

**Базовый синтаксис языка Java. Условные конструкции. Разветвляющиеся алгоритмы**

**Цель работы**

Изучить синтаксис условной конструкции и оператора множественного выбора языка Java для реализаций разветвляющихся алгоритмов (ветвлений) и закрепить их на примере разработки простейших интерактивных консольных Java-приложений.

**Требования**

1) Для каждого вычислительного алгоритма необходимо спроектировать блоксхему решения, которую необходимо поместить в отчёт.

2) На базе спроектированных алгоритмов разработать простейшее консольное

интерактивное приложение с использование архитектурного шаблона проектирования Model-View-Controller, MVC.

3) Создаваемые классы необходимо грамотно разложить по соответствующим

пакетам, которые должны иметь «адекватные» названия и быть вложены в

указанные стартовые пакеты: by.bntu.fitr.nameofstudent.javalabs.lab07.

4) При выполнения задания необходимо по максимуму пытаться разрабатывать

универсальный, масштабируемый, легко поддерживаемый и читаемый код.

5) Также рекомендуется придерживаться Single Responsibility Principle, SRP

(принципа единственной ответственности): у каждого пакета, класса или метода должна быть только одна ответственность (цель), т.е. должна быть только

одна причина изменить в дальнейшем соответствующий блок кода.

6) Если логически не подразумевается или в задании иного не указано, то входными и выходными данными являются вещественные числа (числа с плавающей запятой).

Java Basic Syntax. Conditional Statements. Branching Algorithms

2 | We hope you enjoy working with Java!

7) Все задания необходимо решать используя только базовые операции (простые операторы), определённые над примитивными типами данных в языке

программирования Java, и условные конструкции (т.е. не нужно использовать

циклические конструкции, массивы, строковые данные и операции над ними

и т.д.).

8) В соответствующих компонентах бизнес-логики необходимо предусмотреть

«защиту от дурака».

9) Для генерирования случайных чисел воспользуйтесь методами объекта

класса java.util.Random, а для реализации ввода данных с консоли (терминала) – соответствующими методами объекта класса java.util.Scanner.

10)Программа должна обязательно быть снабжена комментариями, в которых

необходимо указать краткое предназначение программы, номер лабораторной работы и её название, версию программы, ФИО разработчиков, название

бригады (если есть), номер группы и дату разработки. Исходный текст классов

и демонстрационной программы рекомендуется также снабжать поясняющими краткими комментариями.

11)Программа должна быть снабжена дружелюбным и интуитивно понятным интерфейсом для взаимодействия с пользователем. Интерфейс программы и

комментарии в коде должны быть на английском языке.

12) При проверки работоспособности приложения необходимо проверить все

тестовые случаи.

13)При выполнении задания не рекомендуется использовать интегрированные

средства разработки (Integrated Development Environment, IDE). Лучше задействовать любой текстовый редактор и основные компоненты Java (компилятор – javac, утилиту для запуска JVM – java).

14)При разработке программ придерживайтесь соглашений по написанию кода

на Java (Java Code-Convention) !!!

**Основное задание**

1) В молодом возрасте дракон каждый год отращивает по три головы, но после

того, как ему исполнится 200 лет – только по две, а после 300 лет – лишь по

одной. Разработайте программу, которая высчитывала бы, сколько голов и

глаз у дракона, которому N лет. Считать, что при рождении у дракона имеется

уже три головы.

2) Напишите программу «The Greatest», которая определяет какое из трёх (или

четырёх, или пяти и т.д.) введённых пользователем значений наибольшее

(наименьшее). Предусмотреть возможность равенства всех значений.

3) Напишите программу «Mood Sensor» (эмулировать датчика настроения), которая «залазит» в душу пользователя и определяет его настроение в текущий

момент времени. Приложение будет генерировать случайное число, в зависимости от значения которого на экран выводится одно из псевдографических

«лиц», которое и будет отображать настроение пользователя.

4) Напишите программу, которая бы эмулировала игру «Dice» (игра в кости). Суть

игры заключается в броске двух шестигранных кубиков (костей) и подсчёта

общей суммы очков, которые выпали на первой и второй костей.

5) Напишите программу, которая бы определяла, является ли введённая буква

гласной или согласной. Постарайтесь сделать данное задание несколькими

способами (чем больше, тем лучше). К примеру: с использованием базовых

операций, конструкции if-else, конструкции switch и т.д. Ограничения, которые указаны в требованиях, можно игнорировать для данного задания.

**Дополнительное задание**

Заданы три целых числа, которые задают некоторую дату по Грегорианскому

календарю (https://ru.wikipedia.org/wiki/Григорианский\_календарь). Определить дату следующего дня. Запрещается использовать типы стандартной библиотеки языка для работы с датой и временем (можно сделать второй вариант

решения задачи с использование новой библиотеки работы со временем в

Java, которая появилась с JDK 8.0). Также необходимо учесть то, что по Григорианскому календарю (используется в настоящим момент) високосный год

определяется следующим образом:

 годы, кратные 4 – високосные (например, 2008, 2012, 2016);

 годы, кратные 4 и 100 – невисокосные (например, 1700, 1800, 1900);

 годы, кратные 4, 100 и 400 – високосные (например, 1600, 2000, 2400).

**Результаты выполнения основного задания:**

Рисунок 1 – Результат компиляции и запуска программы на языке программирования Java

**Что мы узнали нового в процессе выполнения лабораторной работы:**

*ПРИЛОЖЕНИЕ А*

#### Листинг исходного кода Lab02 на языке программирования Java

}

#### 

**Контрольные вопросы**

1) Для чего в языках программирования нужны управляющие конструкции (операторы)? Чем они управляют?

Operator tells the compiler or interpreter to perform specific mathematical, relational or logical operation and produce final result.

2) Какие существуют в языках программирования фундаментальные типы управляющих конструкций, которые используются при построении любой компьютерной программы? Какие алгоритмы они покрывают?

Arithmetical, relational, bitwise, logical, assignment, miscellaneous.

Liner , cyclic, ramified.

3) Почему необходимо в начале проектировать систему (к примеру, разрабатывать алгоритмы решения задачи), а затем её реализовывать (преступать к

непосредственному написанию кода программы)?

Proper architecture save many efforts ,time and money. Program with proper architecture is easy to expand , change , test, debug, and understand.

4) Опишите основные (базовые) элементы блок-схемы для графического предоставления алгоритмов решения задач?

Start/end block.

Data block. It is used to declaration, initialization.

Action block.

Function involve block.

Condition block.

Cycle blocks.

5) Какие алгоритмы называются разветвляющими и где они применяются?

It’s algorithms have some ways to perform.

Ramified algorithms are used to programs with different processing branches.

6) Какие в языке Java существуют разновидности условных конструкций?

If{

}

Else{}

Switch(n){

Case 1:

Case 2:

Default:}

B=(a==1)?20:22;

7) Как в Java реализуется механизм истинности-ложности?

By using Boolean type.

8) Могут ли различные значения (объекты) выступать в качестве условия (или

условного выражения) истинности-ложности в Java? Какое при этом используется правило?

No.

9) Синтаксис универсальной условной конструкции if-else и её обозначение в

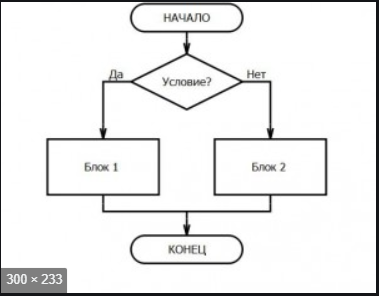
виде блок-схемы? Как она работает? Когда данная конструкция применятся?

If(condition){

…}

Else{

…}



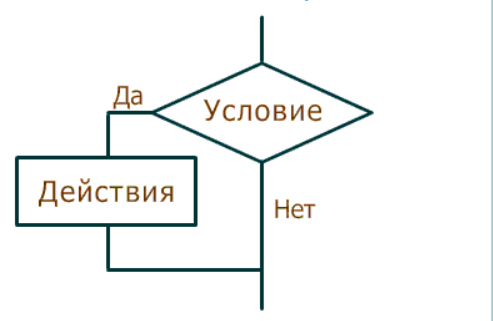
10)Какой логике обычно следуют разработчики при размещении соответствующих инструкций (последовательности операторов) для выполнения после

ключевых слов if и else?

11)Какую разновидность условной конструкции if-else необходимо использовать,

если необходимо выполнить только одно действие при определённом условии (истинности или ложности условия)? Как выглядит она с помощью блоксхемы? Какое используется условие, чтобы данная конструкция была в коде

более читабельна?



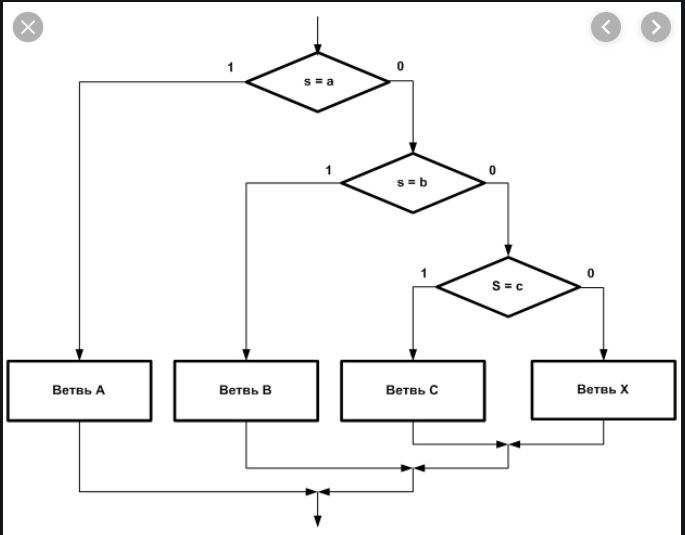
Always use curly braces.

12)Какой в Java используется подход для реализации выполнения кода, в котором

задаётся больше двух условий выполнения? Как выглядит данный подход с

помощью блок-схемы?

Plural “if{}else{}” or switch(choice){case 1: break;}

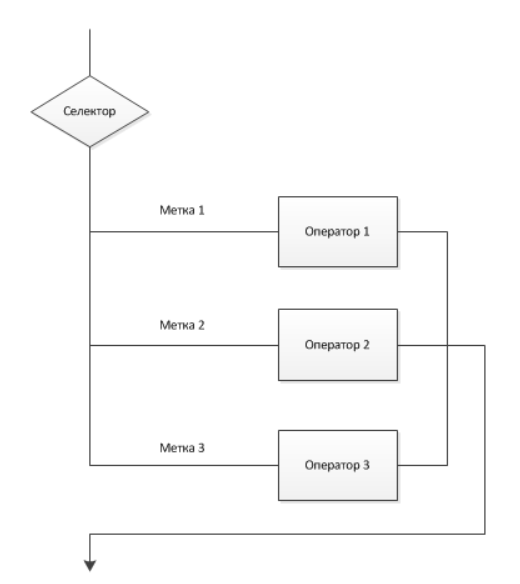


13)Какие есть способы форматирования вышеописанного подхода? Какой из этих способов лучше использовать согласно соглашению?

It depend of current situation in your project. Always use curly braces. It is at ease to use “switch” if we have general selector. If conditionals are different, we are forced to use plural “if”.

14)Каков синтаксис оператора множественного выбора switch-case-default и его

обозначение в виде блок-схемы? Как он работает?



lable: while (true) {

switch(a){

case 0:

Draco dragon = new Draco(*readInt*("Enter age of draco : "));

*exhibit*("Your dragon has " + Draco.*amountOfHeads* + " heads.");

*exhibit*("Your dragon has " + Draco.*amountOfEyes* + " eyes.");

break;

case 1:

break;

case 2:

break;

case 3:

break;

case 4:

break;

case 5:

break;

default:

break lable;

}

}

15)Как происходит каскадное выполнение case-ов в операторе множественного

выбора switch?

Cases are performed one by one while don’t face “break”;

16)Зачем нужна группировка case-ов в условном операторе множественного выбора?

It’s necessary to define bulk of scripts of performing program.

17)Для чего нужно в условном операторе множественного выбора ключевое

слово default и особенность его использования?

If current number in switch doesn’t coincidence with any case, then default is performing.

18)Какие типы данных можно использовать для переменной в конструкции

switch-case-default? Что поменялось в синтаксисе конструкции с выходом различных версий JDK?

Byte,short, int, char and their covers plus enum and String.

19)Чем условная конструкция множественного выбора switch-case-default будет

отличаться от универсальной классической условной конструкции if-else? Когда они применяются в коде?

It is at ease to use “switch” if we have general selector. If conditionals are different, we are forced to use plural “if”.

20)В каких случаях употребление условной (или «тернарной») операции будет

лучше, чем условной конструкции if-else, а в каком случае наоборот?

Ternary operator take less space than “if else”;

It’s used only to assignment.

If actions before condition are complex, than we use “if else”.