**Теория (вопросы)**

1. Что такое оператор ветвления?

Оператор ветвления – это конструкция, в которой в зависимости от истинности некоторого условия, выполняется одно из нескольких действий.

1. Какие типы операторов ветвления вам известны?

if..else, switch..case, :? (тернарный оператор, аналог if..else)

1. Какова синтаксическая структура каждого типа операторов ветвления?

**if\_else:**

if (<условие>) {

<действие>

}

else if (<условие>) {

<действие>

}

else (условие) {

<действие>

}

**switch..case:**

switch (<переменная, которую сравниваем>){

case <переменная1, с которой сравниваем>:

<действие1>;

break;

case <переменная2, с которой сравниваем>:

<действие2>;

break;

default:

<действие>;

break;

Тернарный оператор : ?

<условие> ? <действие, если условие истинно> : <действие, если условие ложно>

1. Что означают ключевые слова “if”, “if else”, “else”. В каком случае их нужно использовать?

if – оператор проверки условия: если условие истинно, то выполняется оператор;

else if - если первое условие if не выполнилось, выполняется переход к else if;

else – если ни одно из предыдущих условий не истинно (if/else..if), то выполняется блок кода в else.

1. Какие типы данных можно использовать в операторе switch..case?

* byte и Byte
* short и Short
* int и Integer
* char и Character
* Enum
* String

1. В каком случае необходимо убрать ключевое слово “break” из оператора switch..case?

break нужно убрать, если нужно объединить условия case, вызывая для них одно и то же действие.

1. Что такое тип Enum?

Enum - это класс перечисления, который помогает ограничить множество допустимых значений для некоторого типа данных.

1. В каком случае лучше использовать оператор ветвления switch..case, в каком тернарный?

Оператор switch..case рекомендуется использовать, когда нам нужно проверить равенство одной переменной со многими значениями. Тернарный оператор рекомендуется использовать только для выражений, содержащих одно условие для проверки и одно действие при каждом исходе.