**Урок 1. Что вообще бывает в коде**

**Цель задания**

Познакомиться с принципом работы программы, научиться его описывать.

**Что нужно сделать**

1. Откройте проект RoadController, запустите файл RoadController.java.
2. Посмотрите, что выведено в консоли.
3. Прочтите код, попытайтесь понять, что он делает.
4. Напишите в ответе, что делает программа.

**Критерии оценки**

«Зачёт» — прислано описание действий, которые выполняет программа.  
«Незачёт» — задание не выполнено.

**Урок 2. Переменные в Java**

**Цель задания**

Научиться читать переменные в коде.

**Что нужно сделать**

1. В коде проекта RoadController обозначьте комментарием строчки, где объявляются переменные.
2. Пришлите проект в архиве.

**Критерии оценки**

«Зачёт» — прислан проект с комментариями в архиве.    
«Незачёт» — задание не выполнено.

**Урок 3. Boolean, операторы сравнения и условные операторы**

**Цель задания**

Научиться работать с переменными boolean и условными операторами if...else.

**Что нужно сделать**  
**Задание №1**

1. Откройте проект **Booleans**.
2. Напишите условные операторы if, проверьте в них количество ингредиентов для каждого рецепта.
3. Убедитесь, что печатаются названия блюд, которые можно приготовить из заданного количества ингредиентов.
4. Попробуйте изменить количество ингредиентов так, чтобы выдавались разные блюда.
5. Пришлите в архиве проект с выполненным заданием.

**Критерии оценки**

«Зачёт» — при достаточном количестве ингредиентов выводятся названия блюд.  
«Незачёт» — задание не выполнено.

**Задание №2**

Напишите программу, которая выводит возрасты трёх людей в порядке по возрастанию.

1. Создайте проект с тремя переменными типа int с именами vasyaAge, katyaAge и mishaAge, обозначающими возрасты трёх человек, как показано в видео.
2. Создайте ещё три переменные типа int с именами min, middle и max.
3. Напишите код, который будет заполнять их, сравнивая возрасты между собой.
4. Выведите значения переменных min, middle и max в консоль. Код должен корректно работать, при изменении значений в изначальных переменных vasyaAge, katyaAge и mishaAge.
5. Пришлите в архиве проект с выполненным заданием.

**Критерии оценки**

«Зачёт» — независимо от возрастов, которые указаны в первых трёх переменных, в консоль возрасты должны выводиться в порядке возрастания: минимальный, средний и максимальный.  
«Незачёт» — задание не выполнено.

|  |
| --- |
|  |

|  |
| --- |
| **Пример результата**  Возрасты в переменных vasyaAge, mashaAge и mishaAge содержат значения 32, 14, 40. Ваша программа должна вывести в консоль:  Minimal age: 14 Middle age: 32 Maximum age: 40 |

**Урок 4. Циклы**

**Цель задания**

Освоить написание циклов for и while.

**Что нужно сделать**

1. Напишите код, используя цикл for, который будет последовательно распечатывать в консоли номера билетов от 200 000 до 210 000 и от 220 000 до 235 000.
2. Распечатайте такие же номера билетов, используя цикл while.
3. Пришлите в архиве проект с заданием.

**Критерии оценки**

«Зачёт» — при запуске проекта в консоли печатаются номера билетов.  
«Незачёт» — задание не выполнено.

**Урок 5. Разбираем сложный код**

**Цель задания**

Закрепить навык чтения кода, научиться понимать и исправлять логику рабочего кода.

**Что нужно сделать**  
**Задание №1**

**1.** Откройте проект RoadController, внесите изменения в код программы:

* Уменьшите высоту пропускного пункта до 3,5 метров.
* Исправьте ошибку: сейчас неверно выставляется цена за грузовые и легковые автомобили в зависимости от веса (weight) транспортного средства.

**2.** Пришлите проект в архиве с изменённым кодом.

**Критерии оценки**

«Зачёт» — прислан проект с исправленным кодом.  
«Незачёт» — задание не выполнено.

**Задание №2**

1. Откройте проект IncomeCalculator. Запустите, посмотрите на выдачу. Прочтите код, попытайтесь понять, что он делает.
2. Опишите в форме ответа общее назначение программы.
3. Напишите к каждой строчке кода комментарий о том, что выполняется в данной строке.

**Критерии оценки**

«Зачёт» — прислан проект с комментариями к коду.  
«Незачёт» — задание не выполнено.

|  |
| --- |
|  |

**Дополнительное задание\***

**1.** Откройте проект IncomeCalculator.  
**2.**Рассчитайте минимальную сумму дохода, при вводе которой программа ответит, что инвестировать можно.

Для этого не обязательно ждать ввод суммы от пользователя. Примените существующие выражения в программе и выведите обратную формулу, удовлетворяющую условию инвестирования.

**3.** Введите значение в консоль в начале метода main():

**public static void** main(String[] args){  
*//расчёт минимальной суммы дохода для инвестирования*  
**while**(**true**){

**4.** Протестируйте получившиеся значения — введите их в программу. Если всё верно, вы получите ответ:

Компания может инвестировать: да

**5.** Введите значение меньше полученного на 1. Если всё верно, вы получите ответ:

Компания может инвестировать: нет

**6.** Пришлите исправленный проект в архиве, в форме ответа напишите сумму, которая у вас получилась.

**Критерии оценки**

«Зачёт» — сумма рассчитана верно.  
«Незачёт» — задание не выполнено.

|  |
| --- |
|  |