

Задание 1

Цель

Научиться писать условные операторы и условия из нескольких выражений, объединённых булевыми операторами.

Что нужно сделать

В готовом проекте напишите код, который будет проверять, достаточно ли ингредиентов для приготовления того или иного блюда.

- Откройте проект Booleans (скачайте прикреплённый архив с проектом).
- Ознакомьтесь с кодом, который написан в классе Main. В нём заданы шесть переменных с количествами разных ингредиентов, а затем показан пример проверки — хватает ли яблок для приготовления яблочного сока.
- Допишите в коде проекта условные операторы if так, чтобы они проверяли количество ингредиентов для каждого рецепта. Если ингредиентов достаточно для приготовления конкретного блюда, в консоль должно выводиться название этого блюда.
- Запустите программу и убедитесь, что печатаются названия блюд, которые можно приготовить из заданного количества ингредиентов.
- Попробуйте изменить количество ингредиентов (значения переменных) так, чтобы выдавались разные блюда. Перед отправкой на проверку убедитесь, что ваш код работает верно.

Рекомендации

В коде есть пример проверки ингредиентов для приготовления яблочного сока. Пишите остальные проверки по аналогии: используйте знак `>=` (больше либо равно), а отдельные условия объединяйте оператором `&&`.

Критерии оценки работы

Принято: при достаточном количестве ингредиентов выводятся названия соответствующих блюд.

На доработку: задание не выполнено или выполнено неточно.

Задание 2

Цель

Научиться писать условия из нескольких выражений и объединять их булевыми операторами.

Что нужно сделать

В новом проекте напишите программу, которая будет выводить заранее заданные возрасты трёх людей в порядке увеличения — от меньшего к большему.

- Создайте проект с именем AgeComparator, создайте в нём класс Main с методом main, а в нём три переменные типа int с именами vasyaAge, katyaAge и mishaAge, обозначающими возрасты трёх человек:

```
int vasyaAge = 29;
int katyaAge = 14;
int mishaAge = 36;
```

- Создайте ещё три переменные типа `int` с именами `min`, `middle` и `max`, в которых в результате выполнения кода должны будут оказаться минимальный, средний и максимальный возрасты соответственно. Пусть изначально они будут равны значению `-1`, это позволит вам проверить, что переменная изменила своё значение в процессе выполнения программы, так как не может быть возраста меньше нуля:

```
int min = -1; // минимальный возраст
int middle = -1; // средний возраст
int max = -1; // максимальный возраст
```

- Напишите код, который будет заполнять эти переменные, сравнивая возрасты между собой. В коде используйте условные операторы `if`, `else` и их комбинацию `else if`, а также при необходимости булевы операторы “И”, “ИЛИ” и “НЕ”.
- Выведите значения переменных `min`, `middle` и `max` в консоль в следующем формате:

```
Minimal age: 14
Middle age: 29
Maximal age: 36
```

- Код должен работать корректно при изменении значений в переменных `vasyaAge`, `katyaAge` и `mishaAge` на любые от 0 до 120.
- Если все возраста различны, код должен работать следующим образом:

```
int vasyaAge = 60;
int katyaAge = 35;
int mishaAge = 15;
Minimal age: 15
Middle age: 35
Maximal age: 60
```

- Если два возраста равны, код должен работать следующим образом:

```
int vasyaAge = 45;
int katyaAge = 17;
int mishaAge = 17;
Minimal age: 17
Middle age: 17
Maximal age: 45
```

- Если равны три возраста, то код должен работать так:

```
int vasyaAge = 55;
int katyaAge = 55;
int mishaAge = 55;
Minimal age: 55
Middle age: 55
Maximal age: 55
```

- Убедитесь, что ваш код работает верно, перед отправкой работы на проверку.

Важно: это специальное задание, над которым нужно подумать. Постарайтесь решить его самостоятельно, пусть это и потребует от вас больше времени и усилий. Если решить его всё же не получится, обратитесь к своему куратору за подсказкой.