



Методические рекомендации по теме

«Составные условия, логические операторы **and**, **or**, **not**»

Цель:

- знакомство с логическими операторами (и, или, не) на языке Python и применение логических операторов.

Задачи:

- расширение кругозора обучающихся в области информатики и программирования;
- знакомство с логическими операторами (и, или, не) на языке Python;
- решение программных прикладных задач с использованием логических операторов на языке Python;
- ранняя профориентация школьников, профессиональная деятельность программиста;
- развитие интеллектуальных способностей, логического и критического мышления

Планируемые результаты

Личностные: обучающиеся получают навыки активной коммуникации в группе, осознанной ориентировки в мире ИТ профессий, постановки собственных образовательных задач и владение первичными навыками деятельностного анализа и критической оценки получаемой информации.

Предметные: обучающиеся получают представления: о логических операторах (и, или, не) на языке Python; об округлении чисел в языке программирования Python; о совмещении условий с помощью логических операторов в языке программирования «Python»; о прикладном использовании логических операторов в программных проектах; о возможностях и особенностях применения логических операторов в практике работы программиста.

Метапредметные: обучающиеся получат возможность владения обще предметными понятиями «совмещение условий»; владение информационно-логическими умениями; умениями самостоятельно планировать пути достижения целей; умениями принятия решений и осуществления осознанного выбора; повысят уровень ИКТ-компетентности.

Материалы к занятию

Приложение 1: Сценарный план видеоролика.

Приложение 2: Домашнее задание и практика.

Приложение 3: Краткие организационно-методические рекомендации по организации работы на занятии.

Ход проведения урока

1. Организационный момент.

Мотивация на учебную деятельность.

Приветствие учащихся, сообщение темы и целей занятия (мы узнаем, что такое «логический оператор» в программировании и как работать с такими операторами на языке Python; научимся использовать логические операторы «и», «или», «не» для объединения условий в коде и сделаем несколько программных проектов, чтобы понять как логические операторы используются программистами).

Проблемная дискуссия по вопросам:

- Как бы вы думаете, что такое «логические операторы»?
- Для чего они могут быть нужны в жизни и программировании?
- Как «логические операторы могут быть связаны с условиями»?
- Какой тип условий можно заменить использованием «логических операторов»?

Итоги дискуссии (обобщаются преподавателем и фиксируются ответы учеников на доске, чтобы вернуться к ним и оценить правильность предположений учеников на этапе рефлексии):

- создавать сложные условия также можно с помощью логических операторов таких как «и», «или», «не»;
- использование логических операторов позволяет оптимизировать громоздкие вложенные условия и делать код более красивым и понятным.

Преподаватель называет ученикам тему и цели урока.

2. Вводный блок.

Тема.

Преподаватель при необходимости останавливая трансляцию, комментируя дополнительно тему занятия)

**см. сцены 1 – 2 (здесь и далее приводится Таблица «Содержание видеоролика». Приложение 1).*

3. Блок повторения.

Блиц-опрос.

Преподаватель предлагает ученикам ответить на **5 вопросов** по предыдущей теме; задания выполняются в сопровождении видеоролика с использованием таймера; ученики выполняют задания, голосуют, обсуждают результаты. Процедура голосования определяется инструкцией **в сцене 3**; учитель должен убедиться, что всем понятна процедура голосования. *Преподаватель может поставить ролик на паузу и обсудить результаты голосования; объяснить правильный ответ руководствуясь материалами предыдущего занятия*

**см. сцены 3 – 7*

4. Теоретический блок.

Логические операторы.

Продолжение демонстрации ролика с дальнейшим обсуждением вопросов:

- Можем ли мы заменить вложенные условия на более простые с использованием логических операторов?
- Какие логические операторы мы рассматривали?
- Опишите особенности работы каждого логического оператора (и, или, не)?

При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу; если ответы на вопросы вызывают у учеников затруднения, преподаватель может вывести нужную сцену ролика на экран для помощи ученикам.

**см. сцены 8 – 13*

5. Блок заданий.

Проект «Двуязычный чат-бот».

К началу демонстрации блока заданий ученики должны занять рабочие места и запустить Python (терминал IDLE) на своих компьютерах.

«Двуязычный чат-бот»: включает **3 практических задания** с таймером; после завершения работы таймера демонстрируется разбор задания. Задания представляют собой небольших программных проекта с использованием логических операторов.

После выполнения задания ученики получают три работающих программных продуктов – программа для определения количества знаков в числе, программа определяющая будние и выходные дни недели, программа для определения повторяющихся чисел в ряду.

На сцене разбора задания преподаватель ставит ролик на паузу и вместе с учениками проводит разбор задания.

**см. сцены 14 – 22*

Блок включает вставку по теории «Объединение логических операторов».

6. Рефлексия. Сообщение домашнего задания.

Завершаем демонстрацией ролика и кратким обобщением материалов занятия. Преподаватель возвращается к зафиксированным в ходе дискуссии в начале урока предположениям учеников и обсуждает насколько их предположения были правильными, делаются выводы.

Преподаватель дает ученикам домашнее задание к следующему занятию (*Приложение 2*).

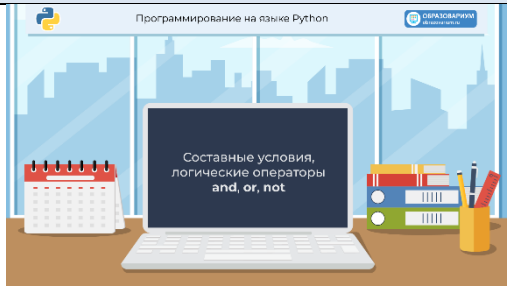
**см. сцена 23*

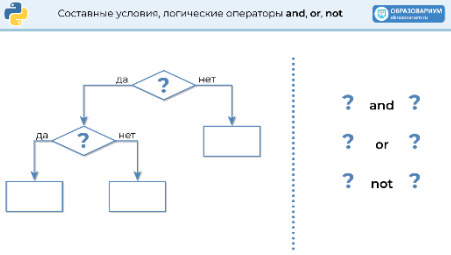
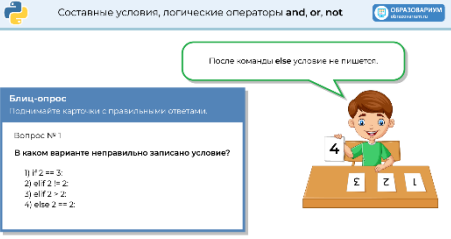
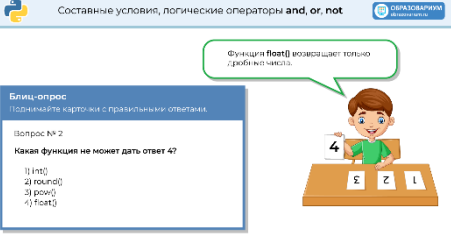
Приложение 1

Сценарный план видеоролика


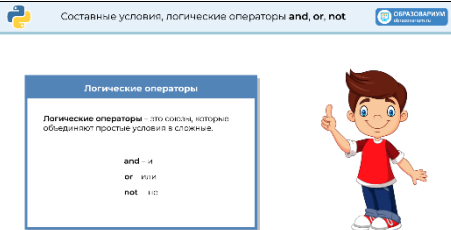

В таблице «Содержание видеоролика» представлен краткий текст из видеоролика, примеры заданий и задач, которые будут демонстрироваться на экране. Учитель при подготовке к уроку может ознакомиться с содержанием видеоролика в текстовом формате, при необходимости распечатать фрагменты текста или примеры заданий и задач для использования в работе с учениками. Распечатанные тексты и задания из таблицы также можно применять в качестве раздаточного материала как на уроке, так и для домашних заданий.

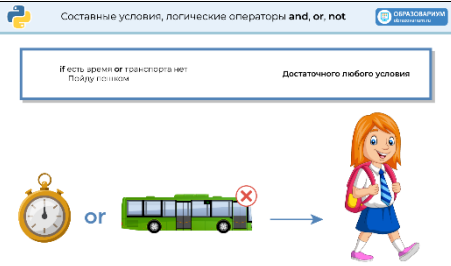
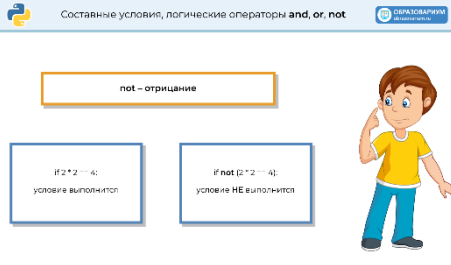
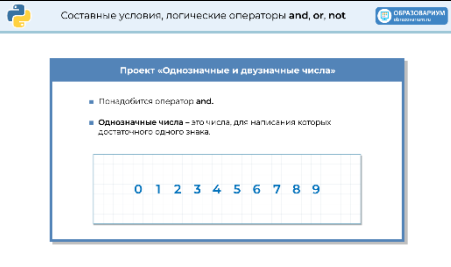
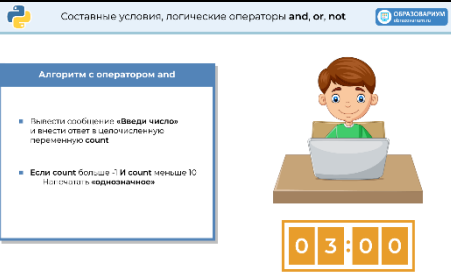
Таблица. Содержание видеоролика

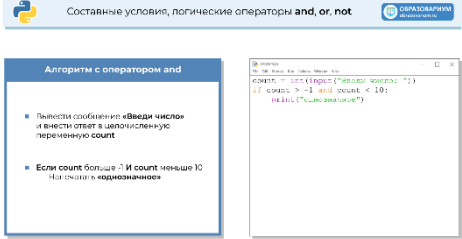
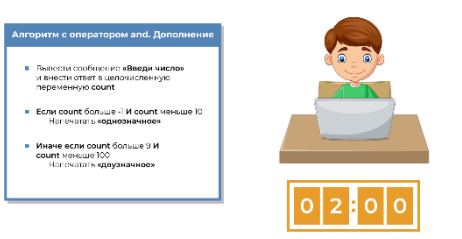
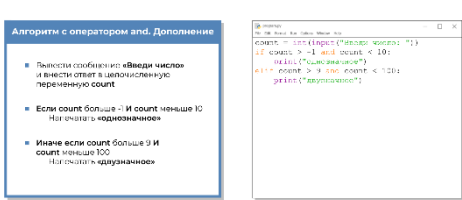
Название блока	Содержание блока и комментарии	Фрагменты из видеоролика	№ сцен
Вводный блок. Мы узнаем	Обозначаем ученикам тему и цели урока. Составные условия, логические операторы and, or, not	 <p>Сцена 1</p>	1 2


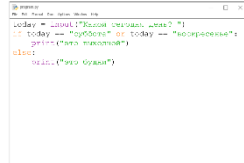
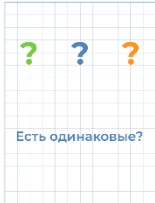

	<p>Для создания сложных, многоуровневых условий используют вложенные условия.</p> <p>Существует еще один способ создания подобных алгоритмов, минуя вложения.</p> <p>Это три известных слова – and, or, not.</p>	 <p>Сцена 1</p>	
<p>Блок повторения.</p> <p>Блиц-опрос</p>	<p><i>Повторение материала предыдущего урока; на столе имеются пронумерованные карточки; после каждого вопроса выбираем ту, номер которой, совпадает с правильным ответом.</i></p> <p>Первый вопрос. В каком варианте неправильно записано условие?</p> <ol style="list-style-type: none"> 1) if 2 == 3: 2) elif 2 != 2: 3) elif 2 > 2: 4) else 2 == 2: <p><i>Ответ 4. После команды else условие не пишется.</i></p>	 <p>Сцена 2</p>	<p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p>
	<p>Второй вопрос. Какая функция не может дать ответ 4?</p> <ol style="list-style-type: none"> 1) int() 2) round() 3) pow() 4) float() <p><i>Ответ 4. Функция float() создает только дробные числа.</i></p>	 <p>Сцена 3</p>	

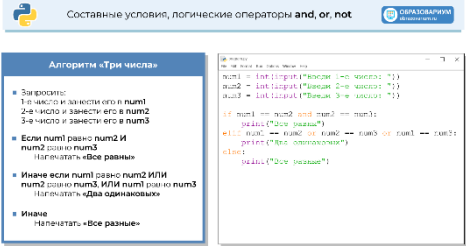

	<p>Третий вопрос. Сколько максимально раз в одном условии могут встретиться if или else?</p> <p>Поднимите карточку с соответствующим числом.</p> <p><i>Ответ 1. if и else в одном условии могут встретиться не больше одного раза.</i></p>	<p>Составные условия, логические операторы and, or, not</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 3 Сколько максимально раз в одном условии могут встретиться if или else?</p> <p>Поднимите карточку с соответствующим числом.</p> <p>Если else в одном условии могут встретиться не больше одного раза.</p> <p>1</p> <p>4 5 2</p>	
	<p>Четвертый вопрос. В каком варианте условие выполнится?</p> <p>1) if $9 == 3 ** 3$; 2) if $4/4 == 4//4$; 3) if $5 * 5 != 25$; 4) if $5 / 0 == 5$?</p> <p><i>Ответ 1. Выполнится условие if $4/4 == 4//4$:</i></p>	<p>Сцена 4</p> <p>Составные условия, логические операторы and, or, not</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 4 В каком варианте условие выполнится?</p> <p>1) $9 == 3 ** 3$; 2) $4 / 4 == 4 // 4$; 3) $5 * 5 != 25$; 4) $5 / 0 == 5$.</p> <p>Выполнится условие: if $4 / 4 == 4 // 4$;</p> <p>2</p> <p>4 5 1</p>	
	<p>Пятый вопрос. В каком варианте есть ошибка?</p> <p>1) int("5") + 5 2) str(5) * 2 3) int("5.0") + 5 4) str(5.0) * 2</p> <p><i>Ответ 3. Запись int("5.0") + 5 выдаст ошибку. Функция int() не может обработать строку "5.0", т. к. в строке записано не целое число.</i></p>	<p>Сцена 5</p> <p>Составные условия, логические операторы and, or, not</p> <p>Блиц-опрос Поднимайте карточки с правильными ответами.</p> <p>Вопрос № 5 В каком варианте есть ошибка?</p> <p>1) int("5") + 5 2) str(5) * 2 3) int("5.0") + 5 4) str(5.0) * 2</p> <p>Запись int("5.0") + 5 выдаст ошибку. Функция int() не может обработать строку "5.0", т. к. в строке записано не целое число.</p> <p>3</p> <p>4 2 1</p>	<p>Сцена 6</p>

<p>Теоретический блок.</p> <p>Логические операторы.</p>	<p><i>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</i></p> <p>Для создания сложного условия в Python необязательно использовать вложенные условия.</p> <p>Есть другой, более простой способ.</p> <p>В этом нам помогут так называемые логические операторы</p>	 <p>Сцена 7</p>  <p>Сцена 8</p>	<p>8</p> <p>9</p> <p>10</p> <p>11</p> <p>12</p> <p>13</p>
	<p>Логические операторы – это союзы, которые объединяют простые условия в сложные.</p> <p>and – и</p> <p>or – или</p> <p>not – не</p>		
	<p><i>Рассмотрим логические операторы на примерах.</i></p> <p>Пример: Оператор and</p> <p>if есть время and погода хорошая</p> <p>пойду пешком</p> <p>Обязательны оба условия</p>	 <p>Сцена 9</p>	

	<p>Пример: Оператор or</p> <p>if есть время or транспорта нет пойду пешком</p> <p>Достаточного любого условия</p>	 <p>Сцена 10</p>	
	<p>Пример: not – отрицание</p> <p>if $2 * 2 == 4$: условие выполнится</p> <p>if not $(2 * 2 == 4)$: условие не выполнится</p>	 <p>Сцена 11</p>	
	<p>Создадим проект на основе оператора and, научим нашу программу определять однозначные числа. Проект «Однозначные и двузначные числа».</p> <p>Для начала вспомним, что это числа, для написания которых достаточно одного знака. То есть от нуля до девяти включительно.</p>	 <p>Сцена 12</p>	
<p>Блок заданий.</p> <p>Практические задания:</p> <p>Задание 1</p> <p>Задание 2</p> <p>Задание 3</p>	<p><i>После окончания дикторского текста запускается таймер на 3 мин.</i></p> <p>Задание 1. Алгоритм с оператором and</p> <ul style="list-style-type: none"> – Вывести сообщение «Введи число:» и внести ответ в целочисленную переменную count. – Если count больше -1 И count меньше 10 – Напечатать «однозначное» 		<p>14</p> <p>15</p> <p>16</p> <p>17</p> <p>18</p> <p>19</p>

	<p>Разбор задания 1. Наша программа может выглядеть таким образом. Обратите внимание, что она игнорирует те числа, которые не являются однозначными.</p> <pre>count = int(input("Введи число: ")) if count > -1 and count < 10: print("однозначное")</pre>	<p>Сцена 13</p>  <p>Сцена 14</p>	
	<p><i>После окончания дикторского текста запускается таймер на 2 мин.</i></p> <p>Задание 1. Алгоритм с оператором and (дополнение) Определяем двузначные числа</p> <ul style="list-style-type: none"> – Иначе если count больше 9 И count меньше 100 – Напечатать «двузначное» 	<p>Сцена 15</p> 	
	<p>Разбор задания 1 (дополнение). Код программы дописывается и может выглядеть таким образом.</p> <pre>elif count > 9 and count < 100: print ("двузначное")</pre>	<p>Сцена 16</p> 	

	<p>После окончания дикторского текста запускается таймер на 4 мин.</p> <p>Задание 2. Алгоритм с оператором or</p> <p>Запросить «Какой сегодня день?» и внести ответ в текстовую переменную today.</p> <p>Если today равно «суббота» ИЛИ today равно «воскресенье»</p> <p>Напечатать «это выходной»</p> <p>Иначе</p> <p>Напечатать «это будни»</p>	<div><div><div>Составные условия, логические операторы and, or, not</div><div>ОБРАЗОВАНИЕ</div></div><div><div>Алгоритм с оператором or</div><div><div>■ Запросить «Какой сегодня день?» и внести ответ в текстовую переменную today</div><div>■ Если today равно «суббота» ИЛИ today равно «воскресенье» Напечатать «это выходной» Иначе Напечатать «это будни»</div></div></div><div><div>04:00</div></div></div> <div>Сцена 17</div>									
	<p>Разбор задания 2.</p> <p>Наша программа может выглядеть таким образом.</p> <pre>today = input("Какой сегодня день? ") if today == "суббота" or today == "воскресенье": print("это выходной") else: print("это будни")</pre>	<div><div><div>Составные условия, логические операторы and, or, not</div><div>ОБРАЗОВАНИЕ</div></div><div><div>Алгоритм с оператором or</div><div><div>■ Запросить «Какой сегодня день?» и внести ответ в текстовую переменную today</div><div>■ Если today равно «суббота» ИЛИ today равно «воскресенье» Напечатать «это выходной» Иначе Напечатать «это будни»</div></div></div><div></div></div> <div>Сцена 18</div>									
<p>«Объединение логических операторов»</p>	<p>При необходимости преподаватель может поставить ролик на паузу и дать дополнительные пояснения по материалу</p> <p>Теперь объединим операторы and и or в одной программе, которая будет запрашивать три числа, а потом определять есть ли среди них одинаковые.</p> <p>Какие в принципе могут быть варианты?</p> <ul style="list-style-type: none">– Все числа одинаковые– Два из них одинаковых– Все числа разные	<div><div><div>Составные условия, логические операторы and, or, not</div><div>ОБРАЗОВАНИЕ</div></div><div><div>Проект «Три числа»</div><table><tr><td>Задача</td><td>Запросить три числа и определить есть ли среди них одинаковые.</td></tr><tr><td>Возможные варианты</td><td><ul style="list-style-type: none">три одинаковых числадва одинаковых числатри разных числа</td></tr><tr><td>Понадобятся операторы</td><td>if – elif – else</td></tr><tr><td>Понадобятся переменные</td><td>num1, num2, num3</td></tr></table><div><p>Есть одинаковые?</p></div></div></div> <div>Сцена 19</div>	Задача	Запросить три числа и определить есть ли среди них одинаковые.	Возможные варианты	<ul style="list-style-type: none">три одинаковых числадва одинаковых числатри разных числа	Понадобятся операторы	if – elif – else	Понадобятся переменные	num1, num2, num3	20
Задача	Запросить три числа и определить есть ли среди них одинаковые.										
Возможные варианты	<ul style="list-style-type: none">три одинаковых числадва одинаковых числатри разных числа										
Понадобятся операторы	if – elif – else										
Понадобятся переменные	num1, num2, num3										
	<p>После окончания дикторского текста запускается таймер на 8 мин.</p> <p>Задание 3. Алгоритм «Три числа»</p> <p>Запросить 1-е число и занести его в num1</p> <p>Запросить 2-е число и занести его в num2</p> <p>Запросить 3-е число и занести его в num3</p> <p>Если num1 равно num2 и num2 равно num3</p> <p>Напечатать «Все равны»</p>	<div><div><div>Составные условия, логические операторы and, or, not</div><div>ОБРАЗОВАНИЕ</div></div><div><div>Алгоритм «Три числа»</div><div><div>■ Запросить: 1-е число и занести его в num1 2-е число и занести его в num2 3-е число и занести его в num3</div><div>■ Если num1 равно num2 И num2 равно num3 Напечатать «Все равны»</div><div>■ Иначе если num1 равно num2 ИЛИ num2 равно num3 ИЛИ num1 равно num3 Напечатать «Два одинаковых»</div><div>■ Иначе Напечатать «Все разные»</div></div></div><div><div>08:00</div></div></div>	21 22								

	<p>Иначе если num1 равно num2 или num2 равно num3 или num1 равно num3</p> <p>Напечатать «Два одинаковых»</p> <p>Иначе</p> <p>Напечатать «Все разные»</p> <p>Разбор задания 3. Наша программа может выглядеть таким образом.</p> <pre>num1 = int(input("Введи первое число: ")) num2 = int(input("Введи второе число: ")) num3 = int(input("Введи третье число: ")) if num1 == num2 and num2 == num3: print("Все равны") elif num1 == num2 or num2 == num3 or num1 == num3: print("Два одинаковых") else: print("Все разные")</pre>	<p>Сцена 20</p>  <p>Сцена 21</p>	
<p>Блок завершения занятия.</p> <p>Рефлексия.</p> <p>Сообщение домашнего задания</p>	<p><i>Завершаем демонстрацией ролика и кратким обобщением материалов занятия.</i></p> <p>Подведем итоги:</p> <p>- узнали, что логические операторы позволяют создавать сложные условия.</p> <p><i>Преподаватель дает ученикам домашнее задание к следующему занятию (Приложение 2).</i></p>	<p>Сцена 22</p> 	23

Приложение 2

Домашнее задание

Придумать и записать свой алгоритм, в котором будут использованы логические операторы.

Практика

Проект «Високосный год»

Запросите номер года. Определите является ли год високосным и выведите соответствующее сообщение на экран.

Напомним, что в соответствии с григорианским календарем, год является високосным, если его номер кратен 4, но не кратен 100, а также если он кратен 400.

Проект «Максимум из трех»

Запросите три числа. Выведите на экран наибольшее из чисел.

Приложение 3

Краткие организационно-методические рекомендации по организации работы на занятии
«Составные условия, логические операторы **and, or, not**».

В начале занятия необходимо вспомнить три команды условного оператора – **if/elif/else**: чем они отличаются друг от друга, последовательность применения, сколько их может быть в одном условии. Особое внимание уделите синтаксису: двоеточия и отступы.

Также нужно вспомнить что из себя представляют **вложенные условия**, для чего они используются и попросить привести примеры, которые ребята должны были сделать в качестве домашнего задания.

Перед просмотром блока повторения из ролика необходимо раздать дидактический материал для выполнения заданий из блока повторение (по 4 пронумерованных карточки)

Во время голосований карточками можно останавливать ролик и вести учет правильных ответов. По окончании блока – отметить тех, у кого наилучший результат.

Далее карточки необходимо собрать.

После теоретического блока можно остановить ролик и попросить привести примеры сложных условий с использованием логических операторов. При формулировках алгоритмов также попросите ребят использовать слова «если», «иначе если», «иначе».

Блок заданий 1 посвящен оператору **and** и состоит из основного проекта и дополнения (двухзначные числа). Если проекты будут сделаны быстро – можно добавить еще условия для определения трехзначных и (или) четырехзначных чисел.

Блок заданий 2 посвящен оператору **or** и состоит только из одного условия. Однако, при наличии времени можно расширить этот проект дополнительными условиями (например: если праздник **ИЛИ** каникулы – тоже выходной). Или придумать собственный вариант.

Задание 3 представляет собой законченный проект и дополнений не предусматривает.