

Е. В. Кривоплясова, В. Ю. Нефёдова, А. В. Прилепина
Оренбургский государственный педагогический университет, Россия

МЕТОДИКА ОБУЧЕНИЯ ОСНОВАМ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PYTHON

Аннотация

В статье рассмотрены основные тенденции изучения языков программирования в школьном курсе информатики. Перечислены допустимые к применению на Едином государственном экзамене по информатике и ИКТ языки программирования (Basic, Pascal, C, Python, алгоритмический язык) и их основные особенности. Проведен анализ использования предложенных языков программирования в учебно-методических комплектах по информатике. Предложена программа элективного курса для обучения основам программирования на языке Python, содержащая пояснительную записку, цели и задачи курса, тематическое планирование, содержание курса, предполагаемые результаты обучения. Рассмотрен методический материал по теме «Программирование ветвящихся алгоритмов» на языке Python.

Ключевые слова: информатика, программирование, язык программирования, программа, алгоритм, учебно-методический комплект, Python.

DOI: 10.32517/2221-1993-2020-19-3-24-30

1. Анализ преподавания раздела «Алгоритмизация и программирование» в школьном курсе информатики

Динамичное развитие информационных технологий в современном мире приводит к появлению все новых языков программирования. Отражение этого процесса мы видим и в школьном образовании, в частности, в расширении перечня языков программирования, допустимых к применению на Едином государственном экзамене по информатике и ИКТ.

Обратившись к КИМ ЕГЭ по информатике и ИКТ [2], мы выделили требования, которые предъявляются к уровню подготовки выпускников в блоке заданий, относящихся к теме «Алгоритмизация и программирование». Необходимы знания: основных конструкций языка программирования, описания и использования

переменной, типов данных, оператора присваивания, операторов ввода и вывода данных, условного оператора, операторов цикла, подпрограмм и рекурсивных функций, массивов и множеств, работы известных и широко используемых алгоритмов, например алгоритма Евклида для вычисления НОД, и многого другого.

Тема «Алгоритмизация и программирование» считается одной из самых сложных в школьном курсе информатики как в теоретическом, так и в практическом плане. На ЕГЭ за задания по этой теме можно получить 15 первичных баллов (задания содержательного раздела «Элементы теории алгоритмов»: 6, 14, 20, 22 (по 1 баллу), 25 (2 балла); раздела «Программирование» — 8, 21 (по 1 баллу), 24 (3 балла), 27 (4 балла)). Задачи данной темы относятся к разным уровням сложности, составляют большую часть экзаменационной работы и, соответственно, приносят наибольшее количество баллов [2].

* Материалы к статье можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_03-2020/

Контактная информация

Кривоплясова Елена Васильевна, канд. пед. наук, доцент, доцент кафедры информатики, физики, методики преподавания информатики и физики, Оренбургский государственный педагогический университет, Россия; *адрес:* 460014, Россия, г. Оренбург, ул. Советская, д. 19; *e-mail:* ewb2001@mail.ru

Нефёдова Виктория Юрьевна, канд. пед. наук, доцент, доцент кафедры информатики, физики, методики преподавания информатики и физики, Оренбургский государственный педагогический университет, Россия; *адрес:* 460014, Россия, г. Оренбург, ул. Советская, д. 19; *e-mail:* vnyefedova@yandex.ru

Прилепина Анна Васильевна, канд. пед. наук, доцент, доцент кафедры информатики, физики, методики преподавания информатики и физики, Оренбургский государственный педагогический университет, Россия; *адрес:* 460014, Россия, г. Оренбург, ул. Советская, д. 19; *e-mail:* apoly@yandex.ru

E. V. Krivoplyasova, V. Yu. Nefyodova, A. V. Prilepina
Orenburg State Pedagogical University, Russia

METHODS OF TEACHING THE BASICS OF PROGRAMMING IN PYTHON

Abstract

The article discusses the main trends in the study of programming languages in the school course of informatics. The programming languages (Basic, Pascal, C, Python, algorithmic language) that are acceptable for use at the Unified State Exam in informatics and ICT and their main features are listed. The analysis of the use of the proposed programming languages in educational and methodological kits on informatics is carried out. The program of the elective course for teaching the basics of programming in Python is proposed. It contains an explanatory note, goals and objectives of the course, thematic planning, course content, expected learning outcomes. The methodical material on the theme "Programming of branching algorithms" in Python is considered.

Keywords: informatics, programming, programming language, program, algorithm, educational and methodological kit, Python.

Традиционно в школьном курсе информатики отводится место на рассмотрение базовых алгоритмических конструкций и основ программирования на одном из языков высокого уровня, который должен быть учебным и универсальным, иметь достаточно простой синтаксис и низкие аппаратно-системные требования. В традициях отечественного образования прочно закрепился язык программирования Pascal, однако в последнее время получили распространение и другие языки. В пробных вариантах заданий по теме «Алгоритмизация и программирование» Единого государственного экзамена выделяют пять языков, а именно: Basic, Pascal, C, Python, алгоритмический язык.

Basic является первым языком программирования, который был предложен для обучения в школьном курсе информатики. Его бесспорными достоинствами были легкий синтаксис и четкие сообщения об ошибках.

Наиболее популярному в учебной практике нашей страны структурному языку программирования Pascal присуща строжайшая типизация, что способствует дисциплинированному программированию, так как в языке максимально исключены различные синтаксические ошибки, а в соответствующих средах программирования присутствует интуитивно понятный интерфейс.

Язык C предназначен в основном для профессионального программирования. В силу сложного синтаксиса он может быть предложен для обучения не всем школьникам, а только наиболее интересующимся информационными технологиями.

Относительно недавно разработанный язык программирования Python имеет лаконичный и в то же время довольно простой и понятный синтаксис, в связи с чем он все чаще предлагается к обучению авторами учебных пособий. Python поддерживает модули и пакеты, поощряя модульность и повторное использование программного кода. Интерпретатор Python и большая стандартная библиотека доступны бесплатно в виде исходных и исполняемых кодов для всех основных платформ и могут свободно распространяться. Язык Python поддерживает несколько парадигм программирования: императивное (процедурный, структурный, модульный подходы), объектно-ориентированное и функциональное программирование.

Алгоритмический язык программирования может быть предложен для решения задач учащимся, не знакомым с другими языками программирования.

Нами были рассмотрены различные учебно-методические комплекты по информатике на предмет выявления рекомендованных к изучению языков программирования.

В УМК Л. Л. Босовой, А. Ю. Босовой [1] программирование рассматривается на базовом уровне в XI классе в разделе «Алгоритмы и элементы программирования». Сначала авторы предлагают повторение рассмотренных ранее сведений о понятии алгоритма, а затем приступают к введению определения массива и операций над массивами, к работе с подпрограммами и рекурсивными алгоритмами. В качестве основополагающего берется язык программирования Pascal.

В УМК И. Г. Семакина и др. для X–XI классов на базовом уровне [5] рассмотрение понятий теории алгоритмов плавно переходит к программированию на языке Pascal. Раздел «Программирование» изучается

в X классе. Преимуществом данного УМК мы считаем рассмотрение символьных, комбинированных и строковых типов данных.

На углубленном уровне обучения информатике Н. Н. Самылкина, И. А. Калинин [4] предлагают к использованию универсальный язык C и допускают возможность самостоятельной теоретической подготовки учащихся с использованием языка программирования Python.

В УМК К. Ю. Полякова, Е. А. Еремина [3] рассматривается язык программирования Python, причем как на базовом, так и на углубленном уровнях обучения. На базовом уровне вводятся представления о языке, типах данных, простейших операциях, переменных, вычислениях и т. д. В учебном пособии для углубленного изучения язык Python представлен более широко.

Анализ указанных УМК позволяет сделать следующие выводы.

Во-первых, в большинстве УМК предлагается рассматривать основы алгоритмизации и программирования на примере нескольких языков программирования, среди которых чаще всего встречается Pascal. Также все большую популярность набирает Python.

Во-вторых, авторский коллектив К. Ю. Полякова, Е. А. Еремина, в отличие от других авторов, помимо Pascal подробно знакомит еще и с Python. Однако материал в УМК этих авторов преподносится на сложном уровне, который рассчитан на учеников с высоким уровнем физико-математических и информационно-технологических знаний.

Исходя из вышеизложенного, нами была признана актуальной задача разработки элективного курса по обучению основам программирования на языке Python для учащихся со средним и повышенным уровнем подготовки, для ребят, которые заинтересованы в знакомстве с альтернативным языком, в повышении уровня своих знаний в программировании.

Разработанный нами элективный курс «Основы программирования на языке Python» можно использовать для учащихся как классов общего профиля, так и классов физико-математического и информационно-технологического профилей.

2. Характеристика элективного курса «Основы программирования на языке Python»

2.1. Пояснительная записка

Программа элективного курса для обучения основам программирования на языке Python разработана для внеурочной деятельности и рассчитана **на учащихся X–XI классов физико-математического и информационно-технологического профилей.**

Курс предполагает определение базовых понятий программирования с дальнейшим применением полученных знаний и навыков при решении задач любого уровня сложности, в том числе представленных в материалах ЕГЭ по информатике и ИКТ. Данный курс помогает развивать школьный курс информатики, осуществляет метапредметную связь с другими дисциплинами (математикой, физикой) и помимо этого является информационной поддержкой выбранного профиля, помогает

школьникам определиться с выбором профессии. При изучении материала курса ученики получают новые знания, обогащают свой жизненный опыт, получают возможность практического применения своих интеллектуальных способностей, развивают логическое и алгоритмическое мышление. Для учащихся любого уровня подготовки дается достаточный объем теоретических и практических знаний в области алгоритмизации и программирования, создаются условия для социального, культурного и профессионального самоопределения, творческой самореализации личности. Изучая программирование, учащиеся лучше понимают суть работы специалистов в данной области, возможности и границы применения компьютеров. Далеко не все будут осваивать профессию программиста, но главным результатом будет то, что ребята освоят азы программирования и узнают, что значит создавать собственные программы.

Определение структуры, содержания и методики преподавания программирования базируется, прежде всего, на следующих компонентах:

- Социальная потребность в наличии учебного курса, предназначенного для реализации обучения.
- Особенность методической составляющей, основанной на активизации самостоятельной деятельности школьника, что способствует формированию навыков экспериментально-исследовательской и информационно-поисковой деятельности (самообучение, самоопределение при постановке задач на этапе разработки проектного задания).
- Приоритетность отбора содержательной компоненты и методических приемов реализации учебного курса, построенных на личностно-ориентированном подходе. Личностно-ориентированный подход в обучении программированию предоставляет возможность каждому ученику изучать материал в собственном темпе относительно своих способностей и потребностей, что способствует эффективному усвоению материала. В рамках данного подхода особую ценность представляет метод проектов, направленный на развитие собственного интеллекта в самостоятельной творческой деятельности школьника, мотивацию к самостоятельной работе по решению интересующей проблемы, приобретение опыта групповой деятельности, что способствует развитию коммуникативных навыков и умений работы в коллективе.

Элективный курс рассчитан на **один учебный год (36 часов)** с расчетом проведения занятий **по 1 часу в неделю**.

2.2. Цели и задачи курса.

Планируемые образовательные результаты

Цели курса:

- формирование у учащихся ИКТ-компетентности;
- формирование знаний и умений решать логические задачи на языке Python;
- развитие интеллекта учащихся и их способностей к логическому и алгоритмическому мышлению;
- подготовка к выбору профессии и дальнейшему изучению программирования.

Задачи курса:

- сформировать у учащихся интерес к изучению программирования;
- дать логически стройное изложение теоретического материала и показать взаимосвязь курса с другими науками;
- познакомить учащихся с базовыми понятиями программирования на языке Python;
- привить умения и навыки использования полученных теоретических сведений для решения прикладных задач;
- подготовить учащихся к успешной сдаче ЕГЭ по информатике и ИКТ.

Требования к уровню освоения учебного курса.

После прохождения курса у учащихся должны сформироваться определенные знания, умения и навыки, которые необходимы для решения задач ЕГЭ по информатике и ИКТ.

Изучение учебного курса будет способствовать формированию следующих личностных, метапредметных и предметных результатов обучения:

личностные:

- бережное отношение к компьютерной технике;
- сформированность мировоззрения, соответствующего уровню развития информационных технологий;
- потребность в саморазвитии в области логического мышления;
- понимание сущности алгоритма в информационных процессах;
- готовность к самостоятельной творческой деятельности;
- готовность и способность вести диалог с другими людьми, сформированность навыков сотрудничества;

метапредметные:

- владение навыками познавательной, учебно-исследовательской и проектной деятельности;
- владение логическими операциями анализа, синтеза, сравнения;
- способность к самостоятельному поиску информации в интернете;
- умение критически оценивать и интерпретировать информацию, получаемую из различных источников;
- умение определять цели деятельности и планировать ее, контролировать и корректировать деятельность;
- умение выбирать стратегию поведения, позволяющую достичь максимального эффекта;

предметные:

в результате изучения элективного курса учащиеся должны:

знать:

- место языка программирования Python среди других языков, структуру программы на языке Python, особенности языка, его достоинства и недостатки;
- основные конструкции языка программирования, величины и их характеристики, математические функции;

- структуры и типы данных, их свойства, характеристики;
- основные алгоритмические конструкции, правила их описания;
- отличия между глобальными, локальными, формальными переменными;
- значение, правила описания и вызова процедур и функций;
- значение, применение и описание рекурсивного алгоритма;

уметь:

- записывать логические и арифметические выражения, разрабатывать программы для обработки числовой и символьной информации;
- разрабатывать программы, содержащие основные алгоритмические конструкции, алгоритмы сортировки и поиска в массивах;
- читать и записывать файлы в заданном формате;
- решать задачи в соответствии с изученным материалом на языке программирования Python и задачи, представленные в КИМ ЕГЭ.

2.3. Формы организации учебных занятий

Занятия включают лекционную и практическую части. Каждое лекционное занятие подкрепляется практическими заданиями.

При проведении занятий используются словесные, наглядные и практические методы обучения.

- Словесные методы включают в себя объяснения, беседы и дискуссии. Важным аспектом при построении занятия должна быть активная работа не только

учителя, но и учеников — они не должны терять связь с учителем, находясь в роли пассивных слушателей.

- Наглядный метод обучения представляет собой демонстрацию применения теории при решении практических задач.
- Практические методы обучения являются самыми результативными, так как включают выполнение упражнений, лабораторных и практических работ, позволяющих определить уровень освоения материала, развить не только самостоятельность, внимательность и смекалку, но и творческие способности учащихся в процессе поиска решения.

2.4. Формы контроля

Контроль знаний учащихся по курсу включает:

- текущий контроль: практические занятия, направленные на составление программ, нахождение ошибок в готовых программах;
- обобщение материала: контрольная работа, включающая решение типовых заданий из КИМ ЕГЭ;
- обобщающий контроль: защита индивидуальных проектов в форме конференции.

Результаты изучения данного курса могут быть проявлены в рамках:

- текущего контроля: беседы и опросы по изучаемым темам;
- зачетного практикума: практическое выполнение обязательных заданий — мини-проектов;
- обобщающего контроля: итоговое задание выполняется посредством создания зачетной программы-проекта.

2.5. Тематическое планирование

№ п/п	Тема урока	Количество часов			Форма контроля
		Всего	Лекции	Практические занятия	
1	Введение в язык программирования Python	1	1	—	Устный опрос
2	Логические выражения, условный оператор, инструкция if	3	1	2	Практическая работа
3	Циклические алгоритмы	5	1	4	Практическая работа
4	Функции	5	1	4	Практическая работа
5	Структурные типы данных. Списки	4	1	3	Практическая работа
6	Символьные строки	4	1	3	Практическая работа
7	Структурные типы данных. Массивы	6	1	5	Практическая работа
8	Файлы	4	1	3	Практическая работа
9	Обобщение материала	3	1	2	Контрольная работа
10	Защита проектов	1	—	1	Конференция
	ИТОГО:	36	9	27	

2.6. Содержание обучения

Тема 1. Введение в язык программирования Python.

Понятие алгоритма и программы. Структура программы. Создание и развитие языка Python, элементы языка, его особенности, достоинства и недостатки. Переменные. Типы данных. Основные операции. Ввод, вывод данных и отладка программы.

Тема 2. Логические выражения, условный оператор, инструкция if.

Определение логического выражения. Логический тип данных и логические операторы. Определение условного оператора, инструкции if и решение задач.

Тема 3. Циклические алгоритмы.

Виды и типы циклов, написание программ и решение задач ЕГЭ с использованием циклов.

Тема 4. Функции.

Функции и их применение при решении задач. Рекурсия и ее применение при решении задач.

Тема 5. Структурные типы данных. Списки.

Структурные типы данных. Списки и их обработка. Линейный поиск. Копирование списков, добавление и удаление элементов. Обработка. Сортировка. Двоичный поиск.

Тема 6. Символьные строки.

Символьные строки. Обработка строк.

Тема 7. Структурные типы данных. Массивы.

Массивы и основные задачи по обработке массивов. Реализация известных алгоритмов по работе с массивами.

Тема 8. Файлы.

Файлы, основные понятия. Открытие, чтение, запись и закрытие файлов.

2.7. Особенности оценивания работ обучающихся в элективном курсе**Критерии оценки усвоения материала.**

В связи с последовательным изложением от самого простого к сложному, необходимо контролировать усвоение каждого раздела для дальнейшей успешной работы. Весь курс разделен на темы, в конце которых учащиеся сдают практические работы. В начале занятий для более плодотворной работы, мотивации и самооценки учащимся раздаются **маршрутные листы**, в которых они сами выставляют баллы по согласованию с учителем. По итогам набранных баллов в конце урока выставляются оценки. Показателями уровня владения материалом являются усвоение теоретических знаний и умение применять полученные знания при решении теоретических и практических задач различного уровня сложности. На лабораторных работах учащимся предлагается для решения несколько задач различных уровней сложности. Для того чтобы преодолеть минимальный порог, необходимо решить 50 % от всей лабораторной работы. Помимо практического зачета учащиеся сдают тестирование по задачам ЕГЭ. Тесты составляются в соответствии со спецификатором, задания берутся из открытого банка заданий ФИПИ, с портала «Решу ЕГЭ». Преимуществом этого портала является то, что ученик, решая задания, накапливает статистику, по результатам которой проще будет устранить пробелы, так как при случайном составлении теста система ставит задания того типа, в котором чаще всего были допущены ошибки. Для того чтобы получить зачет по тестированию, учащийся решает свой вариант полностью, если в ходе решения были допущены ошибки, проводится работа над ошибками с разбором аналогичного задания.

Требования к оформлению проектной работы.

По завершению курса каждый учащийся должен защитить свою проектную работу. Тему этой работы ученик определяет самостоятельно, а затем согласовывает ее с учителем. Основное требование к работе — создание своей программы по решению определенной задачи. Для того чтобы получить зачет, учащийся должен правильно поставить задачу, составить алгоритм по достижению результата, написать программу, ввести и провести отладку программного кода. Минимальный порог зачисления проекта — работа программы для частных случаев.

2.8. Учебно-методическое и материально-техническое обеспечение образовательного процесса**Литературные источники для учителя и ученика.**

1. Поляков К. Ю., Еремин Е. А. Информатика. 10 класс. Углубленный уровень: учебник в 2 ч. Ч. 1. М.: БИНОМ. Лаборатория знаний, 2015. 344 с.

2. Поляков К. Ю., Еремин Е. А. Информатика. 10 класс. Углубленный уровень: учебник в 2 ч. Ч. 2. М.: БИНОМ. Лаборатория знаний, 2015. 304 с.

3. Федоров Д. Ю. Программирование на языке высокого уровня Python. М.: Юрайт, 2019. 126 с.

4. Луцц М. Изучаем Python. 4-е изд. СПб.: Символ-плюс, 2011. 1280 с.

Аппаратное обеспечение, используемое в курсе:

- персональный компьютер и проектор.

С помощью данного оборудования учитель, используя режим трансляции экрана своего компьютера, наглядно демонстрирует особенности создания программ на языке программирования Python.

Программное обеспечение, используемое в курсе:

- программная среда PyCharm.

3. Пример методической разработки урока по курсу «Основы программирования на языке Python»

Тема урока: Логические выражения, условный оператор, инструкция if.

Цели урока:

- познакомить учащихся с логическими выражениями, типами данных, операторами языка программирования Python;
- дать определение условного оператора и инструкции if.

Планируемые образовательные результаты:

- *предметные:* умение решать задачи, используя условный оператор;
- *личностные:* формирование навыков общения; готовность и способность к образованию и самообразованию.
- *метапредметные:* умение контролировать и корректировать учебную деятельность.

Тип урока: открытие нового знания.

Оснащение урока:

- персональный компьютер учителя;
- мультимедийный проектор;
- экран;
- презентация для сопровождения урока*;
- персональные компьютеры учащихся.

Ход урока**I. Мотивационно-целевой этап****Организационный момент.**

Учитель приветствует учащихся, проверяет их готовность к уроку.

* Презентацию можно скачать на сайте ИНФО: http://infojournal.ru/journals/school/school_03-2020/

Определение темы и цели урока.

Учитель. Часто в обыденной жизни человеку приходится делать выбор, что-то отрицать, с чем-то соглашаться. С точки зрения логики ответами на подобные вопросы являются: да (истина) и нет (ложь).

Фактически такой же принцип используется в программировании — если при вычислении значения выражения в результате мы получаем истину или ложь, то выражение является логическим. Соответственно, для подобного рода выражений существует логический тип данных со значениями True и False.

Сегодня на уроке мы рассмотрим логические выражения и работу с ними.

II. Процессуально-познавательный этап**Открытие нового знания.**

Учитель. С основными логическими операторами вы знакомы уже давно, давайте их вспомним (больше, меньше, больше или равно, меньше или равно). Новыми для вас являются обозначения равенства и неравенства: двойное равно (==) и восклицательный знак и равно (!=) соответственно. При этом важно помнить, что одинарное равно обозначает присваивание и не является логическим.

Для образования сложных логических выражений используются операторы объединения: логические И (and) и ИЛИ (or). При их использовании для получения необходимого результата руководствуемся правилами логики.

ОТКРЫТИЕ НОВОГО ЗНАНИЯ

- × обозначение равенства: двойное равно (==)
- × обозначение неравенства: восклицательный знак и равно (!=)
- × для образования сложных логических выражений используем логические И (and) и ИЛИ (or)

Давайте теперь рассмотрим условный оператор.

На примере реальной жизни человека действие этого оператора можно объяснить так: в основном у каждого свой режим дня и расписание дел, которые нужно выполнить в течение дня или недели. Каждое из этих дел может зависеть от определенных обстоятельств, например, если завтра будет мороз, то вы не идете в школу. Или: если человек попадает в ДТП, то он не успевает на встречу, тренировку и т. п.

По такому же принципу строятся нелинейные компьютерные программы: определенная часть кода выполняется только при заданном условии. Для программирования ветвящихся алгоритмов применяются условный оператор и оператор выбора.

Условный оператор имеет следующий формат:

```
if <логическое выражение>:
    <оператор 1>
else:
    <оператор 2>
```

Все структуры, которые относятся к одному блоку, должны иметь равную величину отступа, т. е. одинаковое число пробелов в начале строки.

Также в условной конструкции может отсутствовать оператор **else**, такая структура называется неполным ветвлением. Пример:

```
x = int(input())
if x > 0:
    x = -x
print(x)
```

(Необходимо обратить внимание учащихся на то, что, когда мы вводим число, необходимо указывать тип числа (в данном случае int — целый), в противном случае возникнет ошибка.)

В процессе разработки программы может возникнуть ситуация, в которой после одной проверки следует сделать еще несколько. В таком случае необходимо использовать вложенные условные конструкции. При этом вложенные блоки имеют больший размер отступа. Пример:

```
x = int(input())
y = int(input())
if x > 0:
    if y > 0:
        print("1-я четверть")
    else:
        print("4-я четверть")
else:
    if y > 0:
        print("2-я четверть")
    else:
        print("3-я четверть")
```

Это решение можно записать, используя «каскадную» последовательность, операцией **if ... elif ... else**:

```
x = int(input())
y = int(input())
if x > 0 and y > 0:
    print("1-я")
elif x > 0 and y < 0:
    print("4-я")
elif y > 0:
    print("2-я")
else:
    print("3-я")
```

Обратите внимание на то, что в качестве операторов сравнения используются знаки <, >, <=, >=, == (равно), != (не равно), in (проверка принадлежности объекта диапазону, списку, множеству и т. д.). Результатом вычислений значения логического выражения являются значения True или False, относящиеся к логическому типу (bool).

Часто возникает необходимость проверить несколько условий в одном блоке. В таких случаях допустимо применение логических операторов **or** (или), **and** (и) и **not** (не). Например, фрагмент для проверки того, являются ли числа *a* и *b* четными, будет выглядеть так:

```
(a % 2 == 0) and (b % 2 == 0)
```

ПРОВЕРКА НЕСКОЛЬКИХ УСЛОВИЙ В ОДНОМ БЛОКЕ

Допустимо применение логических операторов `or` (или), `and` (и) и `not` (не).

Например, являются ли числа `a` и `b` чётными:

```
(a % 2 == 0) and (b % 2 == 0).
```

Первичное закрепление материала.

Учитель. Теперь давайте рассмотрим пройденный теоретический материал на практике и проверим, как на самом деле все работает.

Задания.

- 1) С клавиатуры вводится количество денег ученика. Написать программу, которая выдает заключение, хватит ли этих денег для покупки пиццы в буфете.
- 2) Написать программу, которая запрашивает возраст человека и в зависимости от результата выдает сообщение о том, совершеннолетний ли пользователь.
- 3) Написать программу, которая по введенным пяти оценкам за четверть определяет, отличник ли пользователь.
- 4) В забеге на 100 м участвовали четыре спортсмена. Написать программу, которая определяет победителя соревнования, если известно время забега каждого.
- 5) Даны три действительных числа. Возвести в квадрат те из них, значения которых неотрицательны, и в четвертую степень — отрицательные.
- 6) Даны три целых числа. Выведите значение наименьшего из них (использовать `elif`).
- 7) Даны два действительных числа `x` и `y`, не равные друг другу. Меньшее из этих двух чисел заменить третьей частью суммы, а большее — их утроенным произведением.
- 8) Даны три целых числа. Определить, сколько среди них совпадающих.
- 9) Дано трехзначное число `K`. Проверить, будет ли сумма его цифр четным числом.
- 10) Заданы четыре координаты вершин прямоугольника. Определить площадь прямоугольника, расположенного в первой координатной четверти.

III. Рефлексивно-оценочный этап

Подведение итогов урока, рефлексия.

Учитель. Итак, урок подходит к концу, пересаживаемся на свои места, давайте подведем итоги:

- Какие цели мы ставили перед собой? Достигли ли их?
- Что повторили и что узнали нового?
- Возникли ли трудности на уроке?
- Есть ли у вас вопросы по пройденной теме?

4. Заключение

Возможности Python предоставляют учащимся эффективные инструменты для интеллектуального развития и обеспечивают альтернативный подход к решению целого ряда задач по программированию. Для полного использования возможностей этого языка необходимо изучить его структуру, содержание и синтаксис. Изучение Python возможно как в урочное время, так и во внеурочной деятельности. В качестве рекомендаций учителю по обучению основам программирования на Python советуем при разборе формата и выполнения определенного оператора проводить параллель с Pascal (при условии, что ученики ранее изучали этот язык программирования) и выделять синтаксические отличия двух языков.

Список использованных источников

1. Босова Л. Л., Босова А. Ю. Информатика. 10–11 классы. Базовый уровень: методическое пособие. М.: БИНОМ. Лаборатория знаний, 2016. 56 с. <http://methodist.lbz.ru/iiumk/informatics/files/bosova-10-11-bu-met.pdf>
2. Кодификатор требований к уровню подготовки выпускников по информатике для составления контрольных измерительных материалов единого государственного экзамена 2019 года // Федеральный институт педагогических измерений. <http://www.fipi.ru>
3. Поляков К. Ю., Еремин Е. А. Информатика. 10–11 классы. Базовый и углубленный уровни. Примерная рабочая программа. М.: БИНОМ. Лаборатория знаний, 2016. 41 с. <http://methodist.lbz.ru/iiumk/informatics/files/polyakov-10-11-bu-uu-prog.pdf>
4. Самылкина Н. Н., Калинин И. А. Информатика. 10–11 классы. Углубленный уровень: методическое пособие. М.: БИНОМ. Лаборатория знаний, 2016. 168 с. <http://methodist.lbz.ru/iiumk/informatics/files/samykina-kalinin-10-11-uu-met.pdf>
5. Семакин И. Г. Информатика. 10–11 классы. Базовый уровень. Примерная рабочая программа. М.: БИНОМ. Лаборатория знаний, 2016. 33 с. <http://methodist.lbz.ru/iiumk/informatics/files/semakin-10-11-bu-prog.pdf>