

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ГИМНАЗИЯ №66 ПРИМОРСКОГО РАЙОНА САНКТ-ПЕТЕРБУРГА**

ПРИНЯТА на заседании Педагогического совета протокол от 29 августа 2023г №1	УТВЕРЖДЕНА приказом директора ГБОУ гимназии №66 от 29августа 2023 №225
--	--

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«Клуб программистов»**

направление «общеинтеллектуальное»
для обучающихся 9 класса

Составитель:
Калинина М.В.,
учитель информатики

Санкт-Петербург
2023г.

Пояснительная записка

Общая характеристика курса:

Актуальность данной образовательной программы состоит в том, современные профессии становятся все более интеллектоёмкими, требующими развитого логического мышления. Поэтому для подготовки обучающихся к жизни в современном информационном обществе в первую очередь необходимо развивать логическое мышление, способности к анализу и синтезу. Алгоритмическое мышление является необходимой частью научного взгляда на мир. В то же время оно включает и некоторые общие мыслительные навыки, способствует формированию научного мировоззрения, стиля жизни современного человека.

Самостоятельное создание компьютерных программ готовит обучающихся к карьере ученого в компьютерной области или программиста, что поддерживает развитие компьютерного мышления и помогает формированию личности, способной применить базовые понятия программирования в разных аспектах своей жизни.

Основная цель курса «Клуб программистов» – формирование у учащихся навыков операционного и логического стиля мышления, представления о приемах и методах программирования через составление алгоритмов и программ.

В соответствии с поставленной целью можно выделить следующие **задачи: образовательные:**

- способствовать формированию учебно-интеллектуальных умений, приёмов мыслительной деятельности, освоению рациональных способов её осуществления на основе учета индивидуальных особенностей учащихся;
- способствовать формированию активного, самостоятельного, креативного мышления;
- научить основным приемам и методам программирования.

развивающие:

- развивать психические познавательные процессы: мышление, восприятие, память, воображение у учащихся;
- развивать представление учащихся о практическом значении информатики.

воспитательные:

- воспитывать культуру алгоритмического мышления;
- воспитывать у учащихся усидчивость, терпение, трудолюбие.

Направление деятельности программы – модифицированная, относится к программам общеинтеллектуальной направленности.

Возраст обучающихся, на которых рассчитана программа

Программа рассчитана на детей 15-16 лет. Количество детей в одной группе – от 8 человек.

Место курса в учебном плане

Программа курса внеурочной деятельности «Клуб программистов» в 9 классе рассчитана на 26 часов в год, 1 час в неделю.

Содержание программы

Программирование – это наиболее важный раздел курса “Информатика и ИКТ”, изучение которого позволяет решать целый ряд дидактических и педагогических задач. Как и математика, программирование очень хорошо тренирует ум, развивает у человека логическое и комбинаторное мышление. Может быть, не последнюю роль в формировании нового человека XXI века сыграют основы логического и комбинаторного мышления, заложенные в школьные годы на уроках программирования.

Современная система программирования Python выгодно отличается от других языков своей наглядностью, что делает ее простой и доступной для школьников. Бейсик – живой, динамично развивающийся язык. С помощью Python можно создавать приложения практически для любой области современных компьютерных технологий: бизнес-приложения, игры, мультимедиа, базы данных. При этом приложения могут быть как простыми, так и очень сложными, в зависимости от поставленной задачи.

Основу курса составляет изучение азов алгоритмизации и программирования, знакомство учащихся с основными понятиями, используемыми в языках программирования высокого уровня. Большинство заданий встречаются в разных темах для того, чтобы показать возможности решения одной и той же задачи или проблемы различными средствами, обеспечивающими достижение требуемого результата, что в итоге приведет к способности выбирать оптимальное решение данной задачи или проблемы.

Формы деятельности:

Основным формой деятельности в данном курсе является *метод проектов*. Проектная деятельность позволяет развить исследовательские и творческие способности учащихся. Роль учителя состоит в кратком по времени объяснении нового материала и постановке задачи, а затем консультировании учащихся в процессе выполнения практического задания.

Систему программирования Python называют также средой проектирования или просто средой Python. Учащиеся должны научиться создавать пусть не очень сложные, но настоящие программные приложения. Программирование – это лишь один из этапов создания приложения. Есть и другие этапы, которые бывают не менее сложными, трудоемкими и ответственными, чем программирование. В частности, этап разработки интерфейса будущего приложения, с использованием различных элементов управления - кнопок, текстовых: графических полей, полос прокруток и многого другого. Программы на языке Python пишутся по большей части именно для того, чтобы управлять этими элементами. Поэтому программирование вместе с другими этапами создания приложения называют проектированием, а само создаваемое приложение - проектом Python. По этой же причине систему программирования Python называют системой или средой проектирования.

Большое внимание уделяется этапу проектирования задач и разработке информационной модели изучаемого объекта или системы. В основе курса лежит формирование теоретической базы и овладение учащимися конкретными навыками использования компьютерных технологий в сфере деятельности – программировании. В соответствии с этим занятия делятся на теоретическую и практическую части. На теоретической части создаются компьютерные модели и алгоритмы решения задач. В ходе практических работ учащиеся пишут программы и проводят компьютерные эксперименты.

Перечень проектов, выполняемых обучающимися:

- Проект “Калькулятор”.
- Проект-задание “Расположение формы и управляющих элементов”.
- Проект “ Сумма цифр четырехзначного числа ”.

- Проект-задание “Тригонометрический калькулятор”.
- Проект-задание “Треугольник”.
- Проект “Строковый калькулятор”.
- Проект “Проверка знаний”.
- Проект-задание “Тест с выборочным ответом”.
- Проект-задание “Игра Баше”.
- Проект-задание “Поиск большего из двух чисел”.
- Проект “Отметка”.
- Проект “Коды символов”.
- Проект-задание “Факториал”.
- Проект “Таблица умножения”.
- Проект “Количество символов”.
- Проект-задание “Слово-перевертыш”.
- Проект “Построение графика функции”.
- Проект-задание “Нахождение суммы и произведения”.

Виды деятельности:

Организация учебного процесса. Организация учебного процесса строилась в двух взаимосвязанных и взаимодополняющих формах:

- *урочная форма*, в которой учитель объясняет новый материал и консультирует учащихся в процессе выполнения ими практических заданий на компьютере;
- *внеурочная форма*, в которой учащиеся после уроков (дома или в школьном компьютерном классе) выполняют на компьютере практические задания для самостоятельного выполнения.

Методы и форма проведения занятий:

Компьютерный практикум. Разработка каждого проекта реализовывалась в форме выполнения практической работы на компьютере (*компьютерный практикум*). В учебном пособии содержатся подробные указания по построению компьютерных моделей и их реализации в форме проектов на языках программирования и в электронных таблицах.

Кроме разработки проектов под руководством учителя учащимся предлагались *практические задания для самостоятельного выполнения*.

Индивидуализация обучения. Учебно-методический комплекс содержит большое количество заданий разного уровня сложности. Это позволяет учителю построить для каждого учащегося индивидуальную образовательную траекторию.

Методы оценивания:

Текущий контроль уровня усвоения материала осуществлялся по результатам выполнения учащимися практических заданий.

Итоговый контроль реализуется в форме *защиты итоговых проектов*, перечень которых приводится ниже. В начале курса каждому учащемуся было предложено самостоятельно в течение всего времени изучения данного курса разработать проект, реализующий компьютерную модель конкретного объекта, явления или процесса из различных предметных областей. В процессе защиты учащиеся представили проект на языке объектно-ориентированного программирования.

Достижение личностных результатов оценивается на качественном уровне (*без отметки*).

Реализация внеурочной деятельности осуществляется без балльного оценивания результатов освоения курса.

Формы контроля

Предметом контроля и оценки являются образовательные продукты учеников. Качество ученической программы оценивается следующими критериями:

- Последовательность действий при разработке программ: постановка задачи, выбор метода решения, составление алгоритма, составление программы, запись программы в компьютер, отладка программы, тестирование программы.
- «Правила хорошего тона» при разработке программ: читаемость и корректность программ, защита от неправильного ввода, понятия хорошего и плохого «стиля программирования».

Контроль за усвоением качества знаний проводится на трех уровнях:

1-й уровень – воспроизводящий (репродуктивный) – предполагает воспроизведение знаний и способов деятельности. Учащийся воспроизводит учебную информацию, выполняет задания по образцу.

2-й уровень – конструктивный предполагает преобразование имеющихся знаний. Ученик может переносить знания в измененную ситуацию, в которой он видит элементы, аналогичные усвоенным;

3-й уровень – творческий предполагает овладение приемами и способами действия. Ученик осуществляет перенос знаний в незнакомую ситуацию, создает новые нестандартные алгоритмы познавательной деятельности.

Текущий контроль знаний осуществляется по результатам выполнения учащимися практических заданий.

Система оценивания – безотметочная.

Итоговый контроль реализуется в форме защиты собственных программ-проектов учащихся. В процессе защиты учащийся должен представить работающую компьютерную программу, которая решает поставленную перед ним задачу, и обосновать способ ее решения.

Результаты освоения программы внеурочной деятельности:

В результате изучения курса обучающиеся получают дальнейшее развитие личностные, регулятивные, коммуникативные и познавательные универсальные учебные действия, учебная (общая и предметная) и общепользовательская ИКТ-компетентность обучающихся.

Формируются и получают развитие *метапредметные результаты*:

- умение самостоятельно планировать пути достижения целей, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- умения соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата;
- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (далее ИКТ-компетентности);

личностные результаты:

- формирование ответственного отношения к учению;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, творческой и других видов деятельности.
- формирование способности обучающихся к саморазвитию и личностному самоопределению, мотивации к целенаправленной познавательной деятельности с целью приобретения профессиональных навыков в ИТ-сфере;
- способность и готовность к принятию ценностей здорового образа жизни за счет знания основных гигиенических, эргономических и технических условий безопасной эксплуатации средств ИКТ.

предметные результаты:

- умение использовать термины понятий «алгоритм», «данные», «программа» через призму практического опыта в ходе создания программных кодов; понимание различий между употреблением этих терминов в обыденной речи и в информатике;
- умение создавать и выполнять программы для решения несложных алгоритмических задач в среде Python;
- практические навыки создания линейных алгоритмов управления исполнителями;
- умение формально выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов;
- умение создавать и выполнять программы для решения алгоритмических задач в программировании Python.

Выпускник на базовом уровне научится:

- составлять алгоритмы, знать его основных свойств, иллюстрировать их на конкретных примерах алгоритмов; определять возможность применения исполнителя для решения конкретной задачи по системе его команд;
- разрабатывать и записывать на языке программирования типовые алгоритмы;
- объяснять структуру основных алгоритмических конструкций и уметь использовать их для построения алгоритмов.

Выпускник на базовом уровне получит возможность научиться:

- понимать основные принципы объектно-ориентированного программирования (инкапсуляция, полиморфизм, наследование);
- использовать при составлении программы типы данных и операторы (процедуры) для Python;
- описывать основные особенности различных технологий программирования (алгоритмического, объектно-ориентированного).

Содержание программы курса внеурочной деятельности

Раздел 1. Основы объектно-ориентированного программирования (6 часов)

Цели изучения курса «Клуб программистов». Техника безопасности и организация рабочего места.

Объекты: свойства, методы, события. Событийные и общие процедуры. Основные типы данных.

Интегрированная среда разработки систем объектно-ориентированного программирования Python. Визуальное конструирование графического интерфейса. Форма и управляющие элементы.

Переменные. Типы переменных. Объявление переменных в среде Python.

Алгоритм как модель деятельности, формы записи алгоритмов в среде Python. Программа в среде Python. Разработка и исполнение простейших программ в среде Python.

Проект-задание “Расположение формы и управляющих элементов”

Раздел 2. Программирование линейных программ в среде Python (6 часов)

Оператор присваивания, ввод, вывод данных в среде Python. Разработка и исполнение программ с использованием операторов присваивания, ввода, вывода данных в среде Python.

Порядок выполнения операций. Разработка и исполнение программ с использованием операций `div`, `\` в среде Python.

Разработка и исполнение линейных программ в среде Python.

- Проект-задание “Треугольник”
- Проект “Сумма цифр четырехзначного числа”
- Проект “Калькулятор”
- Проект-задание “Тригонометрический калькулятор”

Раздел 3. Программирование ветвлений в среде Python (6 часов)

Разветвляющиеся алгоритмы в среде Python. Условный оператор. Сложные условия в среде Python. Логические отношения и операции. Порядок выполнения операций. Разработка и исполнение разветвляющихся программ с использованием сложных условных операторов в среде Python. Разработка и исполнение разветвляющихся программ с использованием сложных условных операторов в среде Python. Строковые и символьные переменные.

- Проект-задание “Поиск большего из двух чисел”
- Проект “Отметка”
- Проект-задание “Игра Баше”.
- Проект-задание “Слово-перевертыш”.

Раздел 4. Программирование циклов в среде Python (6 часов)

Циклические алгоритмы в среде Python. Оператор цикла в среде Python. Разработка и исполнение программ с использованием циклов в среде Python.

Операторы цикла с предусловием, постусловием, известным числом повторений.

Защипливание программ. Разработка и исполнение с использованием цикла с условием в среде Python. Разработка и исполнение программ с использованием оператора цикла с условием в среде Python.

Цикл с переменной в среде Python. Вложенные циклы в среде Python. Цикл «Пока» в среде Python.

Разработка и исполнение программ с использованием сложных условий в среде Python.

- Проект-задание “Нахождение суммы и произведения”
- Проект-задание “Факториал”
- Проект “Построение графика функции”

- Проект “ Таблица умножения ”

Раздел 5. Создание проектов в среде Python (6 часов)

Разработка и выполнение проектов:

- Проект “Проверка знаний”.
- Проект-задание “Тест с выборочным ответом”.
- Проект “Коды символов”.
- Проект “Количество символов”.
- Проект “Строковый калькулятор”.

Раздел 6. Итоговое повторение. Резерв учебного времени (4 часа)

Литература (информационные ресурсы) для учителя

- Ресурсы сайта Л. Шапошниковой (<http://younglinux.info>), содержащие авторские программы

<http://younglinux.info/python.php> (<http://window.edu.ru/resource/825/76825>)

<http://younglinux.info/algorithm>

учебные пособия

http://younglinux.info/sites/default/files/python_structured_programming.pdf

- Сайт Д.П.Кириенко. Московский институт открытого образования. Школа №179

<http://www.179.ru/~dk/python.html>

- Сайт дистанционная подготовка по информатике

<http://informatics.mccme.ru/>

<http://informatics.mccme.ru/course/view.php?id=95>

- Head First Python. Автор: Paul Barry. Издательство: O'Reilly, год: 2010, Язык: Английский, страниц: 494, формат: pdf, размер: 33 МБ (<http://forcoder.ru/python/head-first-python-1191>)
- Hello World!
Computer Programming for Kids and Other Beginners
Second edition, Warren D. Sande and Carter Sande
<http://helloworldbookblog.com/>
- Python for Kids. A Playful Introduction to Programming
by Jason R. Briggs
2012, 344 pp.
ISBN: 978-1-59327-407-8
Full Color
(<http://www.nostarch.com/pythonforkids>)

Литература (информационные ресурсы) для обучающихся

- Изучаем Python Автор: Марк Лутц Издательство: Символ-Плюс ISBN 978-5-93286-159-2, 978-0-596-15806-4; 2011 г. Переводчик: А. Киселев
- Ресурсы сайта профессора, доктора технических наук, учителя информатики высшей категории, автора учебников по информатике К.Ю. Полякова:

http://kpolyakov.spb.ru/loadstat.php?f=/download/ch10-8_python.pdf

<http://kpolyakov.spb.ru/loadstat.php?f=/download/slides10-8py.zip>

<http://kpolyakov.spb.ru/download/progr1011.doc>

http://kpolyakov.spb.ru/download/infobr_2013-6.pdf

КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

№ п/п	Дата	Тема	Примечание
1. Основы объектно-ориентированного программирования (6 часов)			
1		Цели изучения курса «Программирование в среде Python». Техника безопасности и организация рабочего места.	
2		Понятие алгоритма, свойства алгоритмов. Исполнители алгоритмов, система команд исполнителя. Способы записей алгоритмов. Формальное исполнение алгоритмов.	
3		Объекты: свойства, методы, события. Событийные и общие процедуры. Основные типы данных.	
4		Объектно-ориентированное программирование. Графический интерфейс: форма и управляющие элементы. Событийные процедуры. Проект-задание “Расположение формы и управляющих элементов”.	
5		Переменные. Типы переменных. Объявление переменных в среде Python.	
6		Программа в среде Python. Разработка и исполнение простейших программ в среде Python.	
2. Программирование линейных программ в среде Python (6 часов)			
7		Операторы. Оператор присваивания, ввод, вывод данных в среде Python.	
8		Разработка и исполнение программ с использованием операторов присваивания, ввода, вывода данных в среде Python.	
9		Порядок выполнения операций. Проект-задание “Треугольник”.	
10		Разработка и исполнение программ с использованием операций div, \ в среде Python. Проект “Сумма цифр четырехзначного числа”.	
11		Разработка и исполнение линейных программ в среде Python. Проект “Калькулятор”.	
12		Разработка и исполнение линейных программ в среде Python. Проект-задание “Тригонометрический калькулятор”.	
3. Программирование ветвлений в среде Python (6 часов)			
13		Разветвляющиеся алгоритмы в среде Python. Условный оператор.	
14		Разработка и исполнение разветвляющихся программ с использованием условных операторов в среде Python.	
15		Условный оператор. Проект-задание “Поиск большего из двух чисел”.	

16		Оператор выбора. Проект “Отметка”.	
17		Условный оператор. Проект-задание “Игра Баше”.	
18		Строковые и символьные переменные. Проект-задание “Слово-перевертыш”.	
4. Программирование циклов в среде Python (6 часов)			
19		Циклические алгоритмы в среде Python. Оператор цикла с известным числом повторений в среде Python.	
20		Операторы цикла с предусловием и постусловием. Заикливание программ. Разработка и исполнение с использованием цикла с условием в среде Python.	
21		Циклические алгоритмы в среде Python. Проект-задание “Нахождение суммы и произведения”.	
22		Циклические алгоритмы в среде Python. Проект-задание “Факториал”.	
23		Циклические алгоритмы в среде Python. Проект “Построение графика функции”.	
24		Циклические алгоритмы в среде Python. Проект “Таблица умножения”.	
5. Создание проектов в среде Python (6 часов)			
25		Проект “Проверка знаний”	
26		Проект-задание “Тест с выборочным ответом”	
27		Проект “Коды символов”	
28		Проект “Количество символов”	
29		Проект “Строковый калькулятор”	
30		Защита проекта	
Итоговое повторение. Резерв учебного времени (4 часа)			
31		Итоговое повторение. Резерв учебного времени.	
32		Итоговое повторение. Резерв учебного времени.	
33		Итоговое повторение. Резерв учебного времени.	
34		Итоговое повторение. Резерв учебного времени.	