

**Государственное бюджетное общеобразовательное учреждение
средняя школа №368 с углублённым изучением английского языка
Фрунзенского района Санкт-Петербурга**

ПРИНЯТА
Педагогическим советом
ГБОУ средней школы №368
Протокол № 9
от 25.06.2025 года



УТВЕРЖДЕНА
Директор школы №368

С.Н. Соколова
Приказ № 147-од
от 27.06.2025 года

**Дополнительная
общеразвивающая программа
«Юный программист»**

Возраст обучающихся 14-15 лет
Срок освоения: 36 дней

Разработчик:
Дрягин Георгий Владимирович,
педагог дополнительного образования

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Направленность дополнительной общеразвивающей программы «Юный программист» (далее – программа) - техническая.

Адресат программы – программа предназначена для детей 14-15 лет, проявляющих интерес к техническим дисциплинам, программированию, компьютерному и инженерному творчеству. Изучение основ программирования на примере языка Python позволит детям получить навыки алгоритмического мышления, целеполагания,

Актуальность программы заключается в том, что освоение основ программирования связано с процессом глобальной информатизации с процессом информатизации и необходимостью для каждого человека овладеть новейшими цифровыми технологиями для адаптации в современном обществе и реализации в полной мере своего творческого потенциала. Программирование является одной из отраслей в мире, которая в настоящее время развивается наиболее интенсивно. Программирование как один из основных разделов информатики является не только универсальным инструментом компьютерного моделирования объектов и процессов, но и мощным инструментом в развитии мышления школьников. Необходимость описать средствами языка программирования и стандартных алгоритмических конструкций нестандартные процессы, протекающие в реальном мире, формирует умение определять цели своей деятельности, самостоятельно выстраивать учебную задачу, определять пути и методы ее решения, и таким образом, достигать поставленной цели. В основной школе изучение основ программирования может происходить как в условиях дополнительного образования, так и при изучении основных тем на уроках математики, физики, информатики.

Отличительной особенностью программы является использование бесплатных онлайн сред программирования, что позволит обучающимся свободно практиковаться как на личном компьютере, так и в любых общественных классах с доступом в интернет.

Уровень освоения программы - общекультурный.

Объём и срок освоения программы – 36 часов, 36 дней.

Целью программы является формирование навыков алгоритмического мышления и применения его в программировании.

В ходе реализации программы дополнительного образования решаются следующие **задачи**:

образовательные:

- формирование первичных представлений о компьютерных технологиях, их значении в жизни человека, о профессиях, связанных с программированием;

- формирование первоначальных знаний об алгоритмах и их типах;

развивающие:

- развитие логического и алгоритмического мышления; внимания и абстрактного мышления учащихся;

воспитательные:

- воспитание настойчивости в достижении цели, ответственности, дисциплинированности, трудолюбия;

- формирование умения понимать свои возможности, собственные интересы;

Планируемые результаты:

личностные:

–сформированность познавательных интересов, интеллектуальных и творческих способностей учащихся;

–убежденность в возможности познания природы, в необходимости разумного использования достижений науки и технологий для дальнейшего развития человеческого общества, уважение к творцам науки и техники, отношение к информатике как элементу общечеловеческой культуры.

–повышение уровня самооценки учащимися собственных знаний по предмету.

–активное участие в дискуссии, умение строить логическую цепь рассуждения, уметь подготовиться к выступлению и правильно оформлять проектную работу

формирование информационной и алгоритмической культуры личности

метапредметные:

1) умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учебе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

2) умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;

3) умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;

4) умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;

5) владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

6) умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;

7) умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;

8) умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение;

9) формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

Вопросы обучения школьников основам программирования в научной и научно-методической литературе обсуждаются учеными, учителями и специалистами в области информационных технологий. Рассматривая программирование, как общение с компьютером на языке понятном ему, исследователи поддержали идею обучения детей программированию с любого возраста. В соответствии с требованиями ФГОС ООО изучение предметной области «Математика и информатика» должно обеспечить: осознание значения математики и информатики в повседневной жизни человека; понимание роли информационных процессов в современном мире; формирование представлений о математике как части общечеловеческой культуры, универсальном языке науки, позволяющем описывать и изучать реальные процессы и явления. Программирование на Python сможет помочь реализовать данные требования и подготовить школьников к изучению более сложных программных языков, с которыми они столкнутся в старших классах.

предметные:

1) **математика:** развитие умений работать с учебным математическим текстом

(анализировать, извлекать необходимую информацию), точно и грамотно выражать свои мысли с применением математической терминологии и символики, проводить классификации, логические обоснования, доказательства математических утверждений; решение сюжетных задач разных типов на все арифметические действия; применение способа поиска решения задачи, в котором рассуждение строится от условия к требованию или от требования к условию; овладение системой функциональных понятий, развитие умения использовать функционально графические представления для решения различных математических задач, для описания и анализа реальных зависимостей; распознавание верных и неверных высказываний; формирование представления об основных изучаемых понятиях: информация, алгоритм, модель - и их свойствах; формирование представления об основных изучаемых понятиях: информация, алгоритм, модель - и их свойствах;

2) информатика: формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете.

Изучение предметной области «Естественнонаучные предметы» должно обеспечить: овладение умениями формулировать гипотезы, конструировать, проводить эксперименты, оценивать полученные результаты; овладение умением сопоставлять экспериментальные и теоретические знания с объективными реалиями жизни.

Организационно-педагогические условия реализации программы:

- язык реализации – государственный язык РФ - русский;
- форма обучения – очная;
- особенности реализации – модульная. Программа может быть скорректирована в зависимости от возраста учащихся. Некоторые темы взаимосвязаны со школьным курсом и могут с одной стороны служить пропедевтикой, с другой стороны опираться на него. При работе с робототехникой используются межпредметные связи и пропедевтика следующих научных областей: математика, физика, информатика.

- условия набора и формирования групп – (на обучение принимаются дети в возрасте 14 – 15 лет; специальные условия приема – базовые навыки владения компьютером;

- формы организации и проведения занятий:

Формы организации деятельности: индивидуальная. Оптимальная наполняемость группы составляет 10-11 человек. Режим занятий: продолжительность учебного занятия - 1 час. Занятия проводятся 1 раз в неделю.

формы организации деятельности обучающихся на занятии фронтальная: работа со всеми обучающимися одновременно – беседа, объяснение; групповая: организация работы в малых группах, в т.ч. в парах, для выполнения определенных задач; коллективная: организация творческого взаимодействия между всеми детьми одновременно; индивидуальная: отработка отдельных навыков – индивидуальный подход), подгрупповая (работа в парах и малых группах по 2-3 человека при выполнении творческих заданий, сравнении программных продуктов).

Материально-техническое оснащение

1. Интерактивная панель BM GROUP(3840x2160)
2. Системные блоки, монитор, клавиатура, мышь «Бештау»
3. Стол рабочий второй тип(под мониторы)
4. Стол с подкатной тумбой
5. Кресло для учителя
6. Кресло для ученика

УЧЕБНЫЙ ПЛАН

№ п/п	Название раздела, темы	Количество часов			Формы аттестации	контроля/
		Всего	Теория	Практика		
Раздел 1. Основы синтаксиса и арифметики в Python		9	4	5		
1.1	Вводное занятие, простейшая программа по выводу текста	2	1	1	Собеседование, опрос	
1.2	Арифметические и логические операторы и арифметика в Python	2	1	1	Проверочные вопросы на усвоение материала, Выполнение карточки с заданием	
1.3	Операторы ветвления в Python	5	2	3	Проверочные вопросы на усвоение материала, Выполнение карточки с заданием	
Раздел 2. Списки и циклы		10	4	6		
2.1	Списки в Python	4	2	2	Проверочные вопросы на усвоение материала, Выполнение карточки с заданием	
2.2	Циклы в Python	6	2	4	Проверочные вопросы на усвоение материала, Выполнение карточки с заданием	
Раздел 3. Работа со системами счисления на примере Python		8	3	5		
3.1	Системы счисления	3	1	2	Проверочные вопросы на усвоение материала, Выполнение карточки с заданием	
3.2	Оптимизация работы в Python	5	2	3	Проверочные вопросы на усвоение материала, Выполнение карточки с заданием	
Раздел 4. Работа с функциями в языке Python		8	4	4		
4.1	Функции, их свойства и применение	4	2	2	Проверочные вопросы на усвоение материала, Выполнение карточки с заданием	
4.2	Рекурсивные функции	4	2	2	Проверочные вопросы на усвоение материала, Выполнение карточки с заданием	
Итоговая аттестация		1	0	1	Проверочные вопросы на усвоение материала, Выполнение карточки с заданием	
	Итого	36	14	22		

РАБОЧАЯ ПРОГРАММА

Задачи:

образовательные:

- формирование первичных представлений о программировании, ее значении в жизни человека, о профессиях, связанных с программированием;
- формирование умения программирования в программной среде разработки;
- получение сведений об алгоритмизации, кодировании, конструировании и отладке программ;

развивающие:

- развитие логического и алгоритмического мышления;
- развитие внимания и пространственного воображения учащихся;

воспитательные:

- воспитание настойчивости в достижении цели, ответственности, дисциплинированности, трудолюбия;
- формирование умения понимать свои возможности, собственные интересы;
- развитие любознательности и познавательного интереса; совершенствование умения работать в команде по предложенным инструкциям, навыков общения и сотрудничества.
- воспитание аккуратности, точности, бережного отношения к оборудованию.

Содержание программы

Раздел 1. Основы синтаксиса и арифметики в Python. (9 часов)

Теория: История и современность языка Python, сферы его применения и возможности. Основы работы в средах программирования. Простейшая программа вывода текста. Переменные и присваивание. Математические операции над числами. Операторы равенства и неравенства. Арифметические операции и полном и сокращённом виде Логические операции. Операторы ветвления. Полное и неполное ветвление. Оператор множественного выбора. Вложенные условия

Практика: решение задач по карточкам.

Контроль: ответы на вопросы на занятиях. Итоговая работа

Модуль 2. Списки и циклы (10 часов)

Теория: Списки. Их назначение и применение. Методы присоединения элемента к списку append и extend. Метод count, функция len. Цикл for, его применение для работы с циклами. Цикл while, оператор break. Вложенные циклы.

Практика: решение задач по карточкам.

Контроль: ответы на вопросы на занятиях. Итоговая работа

Модуль 3. Работа со системами счисления на примере Python (8 часов)

Теория: Перевод из десятичной системы счисления в другую. Перевод из любой системы счисления в десятичную. Функции all и any. Работа с файлами. Упрощённая запись действия в цикле.

Практика: Решение задач.

Контроль: ответы на вопросы на занятиях. Итоговая работа

Модуль 4. Работа с функциями в языке Python (8 часов)

Теория: Применение функций. Функции без переменных. Функции от нескольких переменных. Функции без возвращаемого значения. Область видимости функции и

глобальные переменные. Рекурсия и рекурсивные функции. Рекурсивные функции с данными о предыдущих шагах.

Практика: Решение задач.

Контроль: ответы на вопросы на занятиях. Итоговая работа

Итоговая аттестация (1 час): зачет в устной форме

Планируемые результаты

Результаты освоения выражаются в освоении знаний и умений, определенных в программе. Конкретный результат каждого занятия - это робот, выполняющий поставленную задачу благодаря программе, написанной учеником. Проверка проводится как визуально - путем совместного тестирования роботов, так и путем изучения программ, созданных учащимися. Результаты каждого занятия вносятся преподавателем в рейтинговую таблицу. Основной способ итоговой проверки - регулярные зачеты с известным набором пройденных тем. Сдача зачета является обязательной, и последующая пересдача ведется «до победного конца».

Воспитательный результат занятий робототехникой можно считать достигнутым, если учащиеся проявляют стремление к самостоятельной работе, усовершенствованию известных алгоритмов, созданию творческих проектов.

Календарно-тематический план

Раздел/№ занятия	Тема занятия, краткое содержание	Количество часов		Дата планируемая	Дата фактическая
		Очно			
		план	факт		
Раздел 1. Основы синтаксиса и арифметики в Python 1.1.1. Вводное занятие, простейшая программа по выводу текста	Теория: История развития языков программирования, типы языков. Практика: простейшая программа	1	1	1 неделя сентября	
1.1.2. Ввод данных, простейшие арифметические операции	Теория: Ввод данных, простейшие арифметические операции Практика: создание и вывод результатов математических операций	1	1	2 неделя сентября	
1.2.1. Неполный оператор ветвления if, операторы сравнения	Теория: Неполный оператор ветвления if, операторы сравнения Практика: Создание программ с выводом результата при определённом значении выражения	1	1	3 неделя сентября	
1.2.2. Логические операторы, составные условия	Теория: логические операторы Практика: Создание программ с выводом результата при определённом значении выражения	1	1	4 неделя сентября	
1.3.1. Полный оператор ветвления	Практика: Строковый тип данных, функции изменения	1	1	1 неделя октября	

if...else, вложенные условия	строки, использование циклов с предусловием				
1.3.2. Операторы деления, сокращённые записи	Практика: Явное и неявное приведение типов, решение задач	1	1	2 неделя октября	
1.3.3. Оператор множественного выбора if...elif...else	Практика: Решение задач по всему разделу	1	1	3 неделя октября	
1.3.4. Практическая работа по теме "Арифметические операторы и операторы ветвления"	Практика: Решение задач по всему разделу	1	1	4 неделя октября	
1.3.5. Проверочная работа по теме "Арифметические операторы и операторы ветвления"	Практика: проверочная работа	1	1	5 неделя октября	
Раздел 2. Списки и циклы	Практика: Считывание, изменение и вывод списков	1	1	1 неделя ноября	
2.1.1. Списки, их свойства, методы, работа со списками					
2.1.2. Множества, их свойства, методы, работа с множествами	Практика: Методы списков	1	1	2 неделя ноября	
2.1.3. Строки, их свойства, методы, работа со строками	Практика: Циклы	1	1	3 неделя ноября	
2.1.4. Строки, их свойства, методы, работа со строками	Практика: Кортежи и методы работы с ними	1	1	4 неделя ноября	
2.2.1. Индексы и срезы в списках	Практика: Решение задач по всему разделу	1	1	1 неделя декабря	
2.2.2. Цикл for, его использование для работы со списками	Теория: Функциональное программирование, создание собственных функций Практика: Создание простейших функций	1	1	2 неделя декабря	
2.2.3. Цикл while, оператор break	Практика: создание и использование функций одной и нескольких переменных, без возвращаемого значения	1	1	3 неделя декабря	
2.2.4. Вложенные циклы	Практика: создание и использование функций одной и нескольких переменных, с возвращаемого значения	1	1	4 неделя декабря	
2.2.5. Практическая работа по теме "Списки"	Практика: создание и использование функций одной и нескольких переменных, с разными форматами возвращаемого значения	1	1	2 неделя января	

2.2.6. Проверочная работа по теме "Списки"	Теория: история использования модулей, обзор существующих модулей, их подключение	1	1	3 неделя января	
Раздел 3. Парадигмы программирования на примере Python 3.1.1. Системы счисления, перевод из десятичной системы счисления в другую	Модуль math, его функции, использование модуля для решения задач	1	1	4 неделя января	
3.1.2. Системы счисления, перевод из любой системы счисления в десятичную	Модуль cmath, его функции, использование модуля для решения задач	1	1	1 неделя февраля	
3.1.3. Работа с система счисления в Python	Модуль numpy, его функции, использование модуля для решения задач	1	1	2 неделя февраля	
3.2.1. Упрощённая запись действия в цикле	Модуль itertools, его функции, использование модуля для решения задач	1	1	3 неделя февраля	
3.2.2. Решение задач по теме "Системы счисления и оптимизация работы"	Модуль fnmatch, его функции, использование модуля для решения задач	1	1	4 неделя февраля	
3.2.3. Работа с файлами	Теория: история подходов к программированию, функциональный подход	1	1	1 неделя марта	
3.2.4. Проверочная работа по теме "Системы счисления и оптимизация работы"	Теория: Определение чистых функций. Практика: написание чистых функций	1	1	2 неделя марта	
3.2.5. Чистые функции	Теория: Определение функций высшего порядка. Практика: написание функций высшего порядка.	1	1	3 неделя марта	
Раздел 4. Функции на примере Python 4.1.1. Функции, их назначение и синтаксис	Практика: Функция filter. Функция map. Функция zip.	1	1	4 неделя марта	
4.1.2. Функции без возвращаемого значения	Практика: Lambda-функции.	1	1	1 неделя апреля	
4.1.3. Функции от нескольких аргументов	Решение задач с применением знаний модуля.	1	1	2 неделя апреля	
4.2.1. Рекурсивные функции	Теория: Классы и объекты в программировании	1	1	3 неделя апреля	
4.2.2. Рекурсивные функции с данными	Практика: Создание класса с применением инкапсуляции	1	1	4 неделя апреля	

о предыдущих шагах					
4.2.3. Рекурсивные функции, решение задач	Практика: Создание класса-родителя и класса наследника	1	1	1 неделя мая	
4.2.4 Глобальные переменные, их использование в функциях	Практика: Создание класса с применением полиморфизма	1	1	2 неделя мая	
4.2.5. Практическая работа по теме Функции	Практика: Создание класса с неизменяемыми полями.	1	1	3 неделя мая	
	Итоговая работа	1	1	4 неделя мая	

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Учебно-методический комплекс

Учебно-методический компонент для педагога и обучающихся

- использование средств ИКТ на занятиях (презентации, видеофильмы, обучающие
- игры, обучающие компьютерные программы, компьютеры, интерактивная доска, проектор);
- использование дидактического материала (карточки задания, схемы, таблицы, инструкции, практические задания);
- учебники, учебные пособия, журналы, книги;
- тематические подборки теоретического материала, игр, практических заданий;
- Ресурсы сети Internet.

Для учителя:

1. Внеурочная деятельность школьников. Методический конструктор: пособие для учителя. / Д.В. Григорьев, П.В. Степанов. – М.: Просвещение, 2010. – 223 с. – (Стандарты второго поколения).
2. «Программирование: типовые задачи, алгоритмы, методы» / Д.М. Златопольский М.: БИНОМ Лаборатория знаний, 2012.
3. «Задачи по программированию»/ под ред. С. М. Окулова - М.: БИНОМ. Лаборатория знаний, 2014.
4. Акимова И.В., Губанова О.М., Титова Н.В. Изучение элементов логического программирования в рамках факультатива по информатике // Современные проблемы науки и образования. 2018. №4. [Электронный ресурс]. URL: <https://s.science-education.ru/pdf/2018/4/27816.pdf> (дата обращения: 29.05.2024).

Для учащихся:

1. Лутц М. Изучаем python //СПб.: Символ-Плюс. – 2011.
2. Видеозаписи лекций “Основы программирования на Python”: [электронный ресурс] URL: <https://youtube.com/playlist?list=PLLC9rD5jg9GtJanX-LcvuO0sf5LKc08Xd> Дата обращения: 01.11.2021
3. Курс на Stepik “Программирование на Python” [электронный ресурс] URL: <https://stepik.org/course/67/syllabus> Дата обращения: 01.11.2024
4. Курс на Stepik “Python: основы и применение” [электронный ресурс] URL: <https://stepik.org/course/512/syllabus> Дата обращения: 01.11.2024
5. Кринкин К.В. и др. Учебное пособие “Информатика. Введение в Python” / , Кринкин К.В., Берленко Т.А., Заславский М.М., Чайка К.В., Размочаева Н.В. // Издательство СПбГЭТУ “ЛЭТИ”
6. Берленко Т.А. и др. Учебник “Введение в информатику” / Берленко Т.А., Шевская Н.В., Чайка К.В., Заславский М.М., Кринкин К.В. // Издательство СПбГЭТУ “ЛЭТИ”. 8,0 п.л.

7. Самоучитель для начинающих python3 <https://pythonworld.ru/samouchitel-python> (дата обращения 24.05.2024)

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Виды контроля, их периодичность.

Входной – собеседование, опрос – 1 раз на вводном занятии, текущий – выполнение тестовых заданий и упражнений с оцениванием результативности – 1 раз по окончании каждого раздела, итоговое оценивание – решение по задаче из каждого раздела и ответы на теоретические вопросы.

В результате освоения программы обучающиеся должны:

Знать:

- названия базовых типов данных и структур данных языка программирования Python3, а также особенности их хранения в памяти;
- основные встроенные функции языка программирования Python3, способы передачи аргументов, особенности вызова функций;
- основные Функции языка программирования Python3, встраивание модулей в программный код, способы применения модулей.

Уметь:

- выбирать и применять базовые типы данных и структуры данных языка программирования Python3 для решения прикладных задач;
- разрабатывать собственные функции на языке программирования Python3, передавать аргументы в функции, возвращать результат из функции для решения прикладных задач;
- писать программный код на языке программирования Python3, запускать и отлаживать свой код для решения прикладных задач.

Рубежный контроль

Тест №1

1. Составить программу для определения чётности числа.
2. Составить программу для определения, попадает ли число в промежуток [3,20]
3. Составить программу для определения определения возраста человека по дате рождения
4. Составить программу для определения, является ли число одновременно чётным, двузначным и заканчивается не на 4
5. Составить программу для определения, удовлетворяет ли число хотя бы одному из признаков: трёхзначное или двузначное, начинающееся на 7

Тест №2

1. Составить программу для определения количества чётных чисел в списке.
2. Составить программу для определения, гласных букв в строке
3. Составить программу для определения количества чисел, в которых начальная буква совпадает с конечной.
4. Составить программу для определения, количества чисел, в которых только чётные цифры.
5. Составить программу для определения, количества пар соседних чисел в списке, чётность которых различна.

Тест №3

1. Составить программу для перевода из десятичной системы счисления в семеричную.
2. Перевести число 2835 из системы счисления с основанием 40 в десятичную.
3. Упростить запись


```
for x in ListA:
    if x > 10:
        ListB.append(x)
```
4. Упростить запись


```
ListA = [5, 13, 51]
count = 0
for i in range(len(ListA)):
    if ListA[i] > 10:
        count+=1
if count == len(ListA):
    print("Every")
else:
    print("Not every")
```
5. Упростить запись


```
ListA = [5, 13, 51]
count = 0
for i in range(len(ListA)):
    if ListA[i] > 10:
        count+=1
if count>0:
    print("Found")
else:
    print("Not found")
```

Тест №4

1. Составить функцию для определения простоты числа.
2. Составить функцию для определения, попадает ли число в промежуток $[3, 20]$
3. Составить функцию для определения определения возраста человека по дате рождения
4. Составить функцию для определения, является ли число одновременно чётным, двузначным и заканчивается не на 4
5. Составить функцию для определения, удовлетворяет ли введённое число логическому выражению $(x \in P) \vee (x \in Q)$, при том, что на числовой прямой даны два отрезка: $P = [10, 29]$ и $Q = [13, 18]$

Критерий оценок:

Процент правильных ответов	При выполнении от 0% до 50% предлагаемых заданий	При выполнении от 50% до 70% предлагаемы х заданий	При выполнении от 70% до 90% предлагаемых заданий	При выполнении от 90% до 100% предлагаемых заданий
Оценка баллах	2	3	4	5

ОБОРУДОВАНИЕ:

1. Интерактивная панель BM GROUP(3840x2160)
2. Системные блоки, монитор, клавиатура, мышь «Бештау»
3. Стол рабочий второй тип(под мониторы)
4. Стол с подкатной тумбой
5. Кресло для учителя
6. Кресло для ученика

Государственное бюджетное общеобразовательное учреждение
средняя общеобразовательная школа № 368
с углубленным изучением английского языка
Фрунзенского района Санкт-Петербурга

ПРИНЯТА

Педагогическим советом
ГБОУ средней школы №368
Протокол №7 от 05.06.2024

УТВЕРЖДЕНА

Директор школы

С.Н. Соколова
Приказ № 165-од
от 06.06.2024

Календарный учебный график
реализации дополнительной общеразвивающей программы
«Юный программист»
На 2024-2025 учебный год

Дата начала обучения по программе	Дата окончания обучения по программе	Всего учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 сентября 2025	24 мая 2025	36	36	36	1 час в неделю