

**Муниципальное бюджетное общеобразовательное учреждение  
Светлянская средняя общеобразовательная школа  
(МБОУ Светлянская СОШ)**

**ПРИНЯТО:**  
На заседании  
Педагогического совета  
Протокол №11 от 30.08.2024г

**УТВЕРЖДАЮ:**  
Директор МБОУ Светлянской СОШ  
\_\_\_\_\_ О.Л.Боброва  
Приказ № 31/5 од от 30.08.2024 г

**Дополнительная общеразвивающая программа  
технической направленности  
«Основы программирования»  
для учащихся 12-16 лет**

**срок реализации программы 1 год**

**Составитель:  
Боброва Ольга Леонидовна**

с. Светлое  
2024 год

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

<b>Направленность программы</b>	техническая
<b>Уровень программы</b>	базовый
<b>Адресат программы</b>	Программа рассчитана на детей от 12 до 16 лет.
<b>Наполняемость группы</b>	10-12 человек
<b>Объем и срок освоения программы</b>	Программа рассчитана на 72 часа, 36 недель по 2 часа в неделю.
<b>Актуальность программы</b>	<p>Дополнительная общеобразовательная программа «Основы программирования» программы состоит в том, что она составлена с учётом современных потребностей рынка в специалистах в области информационных технологий.</p> <p>В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.</p> <p>Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке программирования. В данной программе выбран язык программирования Python. Выбор этого языка программирования обусловлен тем, что он достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. Python является востребованным языком программирования, он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях разработки.</p>
<b>Отличительные особенности</b>	Основа программы - алгоритмика. Умение алгоритмически мыслить потребуется не только на уроках в школе, но и во многих других учебных дисциплинах и повседневной жизни.
<b>Практическая значимость</b>	Обучение по этой программе дает обучающимся возможность достигать цели посредством использования вычислительной техники и реализовать свои идеи и желания в новом стиле. Это будут основы, базовый уровень, но именно этот базис позволит в будущем качественно и эффективно создавать даже самые сложные программы.
<b>Формы организации образовательного процесса</b>	<p>Форма обучения: очная, с применением дистанционных образовательных технологий.</p> <p>Формы проведения занятий: мастер-класс, исследовательский эксперимент, практическое занятие.</p> <p>Формы организации деятельности: групповая, индивидуально-групповая, индивидуальная, по подгруппам.</p>
<b>Режим занятий</b>	<p>Продолжительность одного занятия для детей в возрасте 12-16 лет – 45 минут.</p> <p>При проведении занятий с использованием компьютерной техники продолжительность занятий 45 минут</p>
<b>Формы и</b>	<b>Личностно-ориентированные технологии:</b> личностно-

<p><b>технологии организации образовательного процесса</b></p>	<p>ориентированное обучение, индивидуальное обучение, педагогика сотрудничества, игровые технологии.  <b>Групповые технологии:</b> групповой опрос; общественный смотр знаний; учебная встреча; нетрадиционные занятия (конференция, путешествие, интегрированные занятия и др.  <b>Здоровьесберегающие технологии</b> - система мер, включающая взаимосвязь и взаимодействие всех факторов образовательной среды, направленная на сохранение здоровья ребенка на всех этапах его обучения и развития.  <b>Методы:</b> словесные, практические, наглядные, репродуктивные.</p>
<p><b>Цель</b></p>	<p>Обучение программировать, писать работающие программы через построение алгоритмов, решение простых задач с использованием языка программирования Python.</p>
<p><b>Задачи</b></p>	<p>1. Изучить основы языка программирования Python;  2. Развить навыков алгоритмического и логического мышления, грамотной разработки программ;  3. Способствовать приобретению навыков поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использование информации при решении задач;  4. Развивать навыки инженерного мышления, умение работать как по предложенным инструкциям, так и находить свои собственные пути решения поставленных задач;  5. Воспитать мотивацию учащихся к изобретательству, созданию собственных программных реализаций.</p>
<p><b>Планируемые результаты</b></p>	<p><b>Предметные:</b></p> <ul style="list-style-type: none"> <li>• умение определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;</li> <li>• формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;</li> <li>• развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;</li> <li>• умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;</li> <li>• навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; умение использовать основные управляющие конструкции объектноориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;</li> <li>• умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать</li> </ul>

результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;

- формирование представлений о математических объектах информатики и об их свойствах, умение оперировать с числами в различных системах счисления, измерять количество информации, представлять информацию в структурированном виде;
- формирование умения соблюдать нормы информационной этики и права.

**Личностные:**

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебноисследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

**Метапредметные:.**

умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;

- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
  - умение критически оценивать правильность решения учебно-исследовательской задачи;
  - умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
  - владение основами самоконтроля, способность к принятию решений;
  - умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий

	(ИКТ-компетенция); <ul style="list-style-type: none"> <li>• умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.»), где от каждого звена зависит конечный результат труда).</li> </ul>
--	---

## УЧЕБНЫЙ ПЛАН

№ п/п	Название раздела, темы	Количество часов				Форма контроля
		Всего	Теория	Практика	Контроль	
1	<b>Основы языка программирования Python.</b>	72				
1.1	Знакомство с языком программирования Python	4	2	2		
1.2	Переменные и типы данных	2	1	1		
1.3	Арифметические операции с числами	3	1	2		
1.4	Условные операторы.	3	1	2		
1.5	Циклы и операторы.	3	1	2		
1.6	Списки.	3	1	2		
1.7	Индексы в списках.	2	1	1		
1.8	Срезы в списках	2	1	1		
1.9	Кортежи.	3	1	2		
1.10	Словари и методы.	3	1	2		
1.11	Множества.	3	1	2		
1.12	Функции.	3	1	2		
1.13	Исключения.	2	1	1		
1.14	Работа с файлами	4	1	3		
1.15	Менеджеры контекста.	2	1	1		
1.16	Модули.	3	1	2		
1.17	Основы объектно-ориентированного программирования в Python	4	2	2		
1.18	Наследование.	3	1	2		
1.19	Инкапсуляция.	3	1	2		
1.20	Полиморфизм.	3	1	2		
1.21	Конструкторы в классах.	2	1	1		
1.22	Переопределение методов в классах	2	1	1		
1.23	Декораторы функций	2	1	1		

1.24	Черепашья графика (turtle)	5	1	4		
1.25	Итоговое занятие	3	1		2	Защита мини проекта

## СОДЕРЖАНИЕ РАБОЧЕЙ ПРОГРАММЫ.

### 1. Основы языка программирования Python.

**1.1. Знакомство с языком программирования Python.** Знакомство с языком программирования Python. Используется презентация «Знакомство с языком программирования Python».

#### 1.2. Переменные и типы данных.

Изучение переменных и типов данных в языке программирования Python. Создание программ с использованием переменных. Используется презентация «Переменные и типы данных Python».

#### 1.3. Арифметические операции с числами.

Изучение арифметических операций с числами в языке программирования Python (сложение, вычитание, умножение, деление). Создание программ с использованием арифметических операций. Используется презентация «Арифметические операции с числами Python». 40 минут

#### 1.4. Условные операторы.

Изучение условного оператора if в языке программирования Python. Создание программ с использованием условного оператора if. Используется презентация «Условные операторы Python».

#### 1.5. Циклы и операторы.

Изучение циклов for и while, а также операторов break, continue и else в языке программирования Python. Создание программ с использованием циклов и операторов. Используется презентация «Циклы и операторы Python»

#### 1.6. Списки.

Изучение списков в языке программирования Python. Создание программ с использованием списка list. Используется презентация «Списки Python».

#### 1.7 Индексы в списках

Изучение индекса в списках в языке программирования Python. Создание программ с использованием индекса в списках. Используется презентация «Индексы в списках Python».

#### 1.8 Срезы в списках.

Изучение среза в списках в языке программирования Python. Создание программ с использованием среза в списках. Используется презентация «Срезы в списках Python».

#### 1.9. Кортежи.

Изучение кортежа в языке программирования Python. Создание программ с использованием кортежа (tuple). Используется презентация «Кортежи Python».

#### 1.10 Словари и методы

Изучение словаря и его методов в языке программирования Python. Создание программ с использованием словаря dict. Используется презентация «Словари и методы Python».

#### 1.11. Множества.

Изучение множеств в языке программирования Python. Создание программ с использованием множеств через методы set и frozenset. Используется презентация «Множества Python».

#### 1.12 Функции.

Изучение функций в языке программирования Python. Создание программ с использованием функций def, lambda и return. Используется презентация «Функции Python».

### **1.13. Исключения.**

Изучение исключений в языке программирования Python. Создание программ с использованием конструкций try и except. Используется презентация «Исключения Python».

### **1.14 Работа с файлами**

Изучение работы с файлами в языке программирования Python. Создание программ, в которых мы научимся открывать файлы, записывать в них какую-либо информацию, а также читать информацию с какого-либо файла. Используется презентация «Работа с файлами Python».

### **1.15. Менеджеры контекста.**

Изучение менеджеров контекста в языке программирования Python. Создание программ с использованием конструкций with и as. Используется презентация «Менеджеры контекста Python».

### **1.16. Модули.**

Изучение модулей в языке программирования Python. Создание программ с использованием конструкций import и from. Используется презентация «Модули Python».

### **1.17. Основы объектно-ориентированного программирования в Python.**

Изучение основ объектно-ориентированного программирования (ООП) в языке программирования Python. Создание программ с использованием классов и объектов, а также создание методов и полей для основного класса. Используется презентация «Основы ООП Python».

### **1.18. Наследование.**

Изучение концепции «Наследование» объектно-ориентированного программирования (ООП) в языке программирования Python. Создание программ с использованием данной концепции. Используется презентация «Наследование ООП Python».

### **1.19. Инкапсуляция.**

Изучение концепции «Инкапсуляция» объектно-ориентированного программирования (ООП) в языке программирования Python. Создание программ с использованием данной концепции. Используется презентация «Инкапсуляция ООП Python».

### **1.20. Полиморфизм.**

Изучение концепции «Полиморфизм» объектно-ориентированного программирования (ООП) в языке программирования Python. Создание программ с использованием данной концепции. Используется презентация «Полиморфизм ООП Python».

### **1.21. Конструкторы в классах.**

Изучение конструктора в классе в языке программирования Python. Создание программ с использованием конструктора в классе. Используется презентация «Конструкторы в классах Python».

### **1.22 Декораторы функций**

Изучение конструкции декораторов в языке программирование Python.

Создание программ с использованием декораторов. Используется презентация «Декораторы Python»

### **1.23. Переопределение методов в классах.**

Изучение переопределения метода в классе в языке программирования Python. Создание программ с использованием переопределения метода в классе. Используется презентация «Переопределение методов в классах Python».

### **1.24. Черепашья графика (turtle).**

Изучение «черепашьей графики» (turtle) в языке программирование Python. Создание программ с использованием «черепашьей графики» (turtle). Используется презентация «Turtle graphics Python».

### **1.25. Итоговое занятие.**

**РАБОЧАЯ ПРОГРАММА ВОСПИТАНИЯ,  
КАЛЕНДАРНЫЙ ПЛАН ВОСПИТАТЕЛЬНОЙ РАБОТЫ**

Разделы Рабочей Программы воспитания	
1.1. Традиционные мероприятия 1.2. Коллективно-творческая деятельность 1.3. Социальная активность учащихся	1.4. Профориентационная работа 1.5. Работа с родителями учащихся 1.6. Воспитание медиа-сопровождением

Выбор разделов и мероприятий в календарном плане воспитательной работы программы «Основы проектирования» соответствует ее направленности, возрастным особенностям и деятельности.

<i>Месяц</i>	<i>Раздел</i>	<i>Часы</i>	<i>Мероприятие</i>	<i>Цель, задачи</i>	<i>Мониторинг</i>
сентябрь	1.5	2	Родительское собрание	Знакомство с содержанием ДООП	Аналитическая справка
октябрь	1.6	2	Урок цифры	Выявление одаренных детей, развитие творческих способностей	Количество участников, публикация
ноябрь	1.5	2	Мастер-класс родители и дети	Объединение детей и взрослых для совместной деятельности. Знакомство с экспериментальной деятельностью	Количество участников, публикация
декабрь	1.1	2	Новогодний калейдоскоп	Выявление, поддержка, развитие творческих способностей талантливых детей	Информационная справка
январь	1.4	2	Экскурсия в СЮТ г Воткинска	Духовно-нравственное воспитание детей, профориентация	Аналитическая справка
март	1.1	2	Видеоткрытка к 8 Марта	Популяризация творчества детей. Выявление, поддержка, развитие творческих способностей талантливых детей	Количество участников, публикация
Апрель-май	1.5	2	Родительское собрание – подведение итогов, отчет	Познакомить родителей с Достижениями детей в учебной деятельности за год. Подведение итогов совместного воспитательно-образовательного процесса как средства всестороннего развития	Аналитическая справка
ИТОГО:		14			

**КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК**

К – комплектование групп, О – каникулярное время, П – промежуточная аттестация, Р – резервное время, У – учебные занятия

*1 год обучения*

Сентябрь					Октябрь				Ноябрь				Декабрь				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
К	К	У	У	У	У	У	У	У	У	У	У	У	У	У	У	П	О

Январь			Февраль				Март					Апрель				
	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
О	У	У	У	У	У	У	У	У	У	У	У	У	У	У	У	У

Май				Июнь			
34	35	36	37	38	39	40	41
У	У	У	У	П	О	О	О

1 полугодие	15 недель : с 15 сентября по 30 декабря
каникулы	с 31 декабря по 10 января
2 полугодие	21 неделя: с 11 января по 6 июня

## УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

### **Учебные и вспомогательные помещения:**

1. Учебный кабинет, соответствующий требованиям СанПиН, оборудованный ученическими столами и стульями, столом для учителя.

### **Технические средства обучения:**

1. Ноутбуки с выходом в Интернет.
2. Проектор
3. Экран

### **Оборудование:**

- рабочее место обучающихся: персональный компьютер (ноутбук) с операционной системой Windows, Linux с программным обеспечением для языка программирования Python

### **Дидактический материал:**

1. Тестовые задания.
2. Инструкции для проведения практических работ.

## ФОРМЫ АТЕСТАЦИИ

Предметом диагностики и контроля в курсе «Основы программирования» являются образовательные продукты учащихся: созданные блок-схемы, программы.

Качество образовательной продукции оценивается по следующим параметрам:

алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;

программа должна выполнять поставленные задачи;

по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).

Проверка достигаемых учащимися результатов производится в следующих формах:

- } текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;
- } текущая диагностика и оценка учителем деятельности школьников;
- } итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальной образовательной программой освоения курса.

Практическим результатом обучения воспитанников является также самостоятельно выполненный проект, включающий в себя не только компьютерную программу, но и описание алгоритма её работы, контрольные тесты для её проверки, описание используемых языковых конструкций и оригинальных приёмов программирования, область практического применения и другую документацию, отражающую работу над созданием проекта.

Темы проектов выбираются учащимися как самостоятельно, так и из числа предложенных преподавателем, но всегда обсуждается в группе.

Необходимо чтобы программа имела практическое применение и была интересна не только авторам проекта.

Обеспечивается возможность презентации проекта на школьной научно-практической конференции.

#### **Примерные требования при защите проекта**

Учащиеся должны (по желанию) продемонстрировать самостоятельно разработанную программу, показать владение техническими, программными средствами разработки проекта, знание основ программирования, уметь ответить на вопросы по содержанию.

По техническим средствам:

- какие технические средства были задействованы при создании проекта;
- объяснить минимальные требования к программному и аппаратному обеспечению для работы программы.

По технологии разработки проект:

- охарактеризовать программные средства, используемые при выполнении проекта;
- объяснить технологические приемы разработки;
- наличие документации;
- рекомендации по работе с программой.

По разработке программы:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- код должен быть «читаемым» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.);
- программа должна выполнять поставленные задачи, и правильно работать; - наличие собственных конструктивных решений;
- знание основ программирования и знание технология программирования;
- наличие дружественного интерфейса.

По содержанию проекта:

- актуальность работы,
- глубина и логика раскрытия темы,
- форма подачи информации,
- заинтересованность в проведении исследования,
- области применения разработки, - источники информации.

При подведении итогов учитывается не только результаты диагностики и качество выполнения учебных заданий, но и личные достижения учащихся.

**«Информационная карта освоения образовательной программы»**

№	Ф.И	Критерии наблюдения						Общий суммарный балл	Уровень усвоения
		А	Б	В	Г	Д	Е		

Критерии: «А» Социальная активность обучающегося;  
 «Б» Коммуникабельность, умение работать в коллективе;  
 «В» Владение навыками алгоритмизации и знание основных конструкций языка программирования;  
 «Г» Навыки анализа, синтеза и культуры представления информации;  
 «Д» Умение работать на персональном компьютере в среде программирования;  
 «Е» Владение технологией программирования;  
 «Ж» Уровень самостоятельности;  
 По каждому критерию выставляется: от 0 до 5 баллов.  
 Общий уровень определяется суммированием баллов:  
 - низкий уровень: от 7 до 20 баллов;  
 - средний уровень: от 21 до 27 баллов;  
 - высокий уровень: от 28 до 35 баллов.

### МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

1. Методическое пособие для учителя к завершённой предметной линии учебников «Информатика» для 10–11 классов общеобразовательных учреждений (углублённый уровень) А в т о р ы: К.Ю. Поляков, Е. А. Еремин. БИНОМ. Лаборатория знаний / Автор-составитель: М. Н. Бородин.—Эл. изд.—М. : БИНОМ. Лаборатория знаний, 2013 ISBN 978-5-9963-1344-

2. «Здоровье и безопасность детей в мире компьютерных технологий и Интернета» Электронное методическое пособие. Партнёрство в образовании. Проект «Академия учителей»

### СПИСОК ЛИТЕРАТУРЫ.

#### 1. Интернет ресурсы:

- <https://pythonworld.ru/> - Язык программирования Python для начинающих.
- <https://itproger.com/course/python> - Уроки Python для начинающих.
- 2. Материалы и презентации к урокам в LMS Яндекс.Лицея.
- 3. Сайт pythonworld.ru — «Python для начинающих».
- 4. Сайт pythontutor.ru — «Питонтьютор».
- 5. Домашняя страница Python [www.python.org](http://www.python.org) . Справочные материалы, официальная документация;
- 6. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info> ;
- 7. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info> ;
- 8. Сайт проекта Open Book Project [openbookproject.net](http://openbookproject.net) содержит серию практических примеров на Python Криса Мейерса;
- 9. Интерактивный Python. Trinket <https://trinket.io/python> ;
- 10. Интерактивный учебник языка Питон <http://pythontutor.ru/> ;
- 11. Python 3 для начинающих <https://pythonworld.ru/> ;
- 12. Python GUI Programming (Tkinter) [https://www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm) <http://www.tkdocs.com/tutorial/> и <http://effbot.org/tkinterbook/> ;
- 13. Python IDE for beginners <http://thonny.org/> ;

14. Бесплатный, простой и профессиональный инструмент для изучения программирования с помощью Python <https://www.jetbrains.com/pycharm-edu/> ; 13. 5. Язык Python (Электронный ресурс) <http://kpolyakov.spb.ru/school/probook/python.htm>.

2. Учебники:

А. Г. Юркин. Задачник по программированию. СПб: Питер, 2021 год.

М.Э. Абрамян. Электронный задачник по программированию. Ростов- на –Дону, 2019 год