

Государственное бюджетное общеобразовательное учреждение лицей № 373  
Московского района Санкт-Петербурга «Экономический лицей»

ПРИНЯТА

решением Педагогического совета  
Государственного бюджетного  
общеобразовательного учреждения  
лицей № 373 Московского района  
Санкт-Петербурга «Экономический  
лицей», протокол от 30.08.2023 г. № 1

УТВЕРЖДЕНА

приказом по Государственному бюджетному  
общеобразовательному учреждению лицей  
№ 373 Московского района Санкт-Петербурга  
«Экономический лицей» от 30.08.2023 № 120-од



**РАБОЧАЯ ПРОГРАММА**  
**курса внеурочной деятельности**  
**«Программирование на языке Python»**  
**6 класс**

Учитель-составитель:

Медведева Людмила Анатольевна,  
учитель информатики ГБОУ лицей № 373  
Московского района Санкт-Петербурга

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Рабочая программа курса внеурочной деятельности «Программирование на языке Python» разработана на основе:

- Федерального государственного образовательного стандарта основного общего образования, утвержденного приказом Министерства просвещения Российской Федерации от 31.05.2021 № 287 (с изменениями и дополнениями);
- Федеральной образовательной программы основного общего образования, утвержденной приказом Министерства просвещения Российской Федерации от 18.05.2023 №370;
- Положения «О рабочей программе учебного предмета, курса государственного бюджетного общеобразовательного учреждения линей № 373 Московского района Санкт-Петербурга «Экономический лицей».

Настоящая рабочая программа является составной частью основной образовательной программы основного общего образования ГБОУ лицей № 373 Московского района Санкт-Петербурга (содержательный раздел).

«Программирование на языке Python» является прикладным курсом внеурочной деятельности учащихся, реализующим интересы обучающихся 6 класса в сфере изучения информационных технологий и языков программирования.

Курс рассчитан на 34 часа для учащихся 6 классов и реализуется в течение одного года.

При реализации данной программы могут применяться электронные образовательные ресурсы и дистанционные образовательные технологии.

Электронный ресурс:

Курс «Python для начинающих» на платформе Stepik:

<https://stepik.org/course/90478/syllabus>

Направление программы – общеинтеллектуальное

Актуальность курса связана с бурным развитием IT-технологий. Понятно, что чем раньше учащийся начнет овладевать навыками программирования, тем больший запас знаний и технологий он получит к моменту выбора основного рода деятельности. Даже если в будущем карьерный путь ребенка не будет связан с программированием, умение разбираться в сложных алгоритмических системах и взаимодействовать с новыми технологиями ему пригодится в любой сфере, ведь цифровые технологии используются повсеместно.

Внеурочный формат занятий позволяет снять с ребенка давление школьных оценок. Во время самостоятельной работы над микро-проектами ребенок обязательно сделает множество ошибок, но при правильной поддержке со стороны преподавателя работа над ошибками позволит почувствовать их ценность. Ведь именно поиск ошибок и последовательное их исправление позволяет улучшать мир вокруг нас и настраивать сложные системы.

**Цель реализации программы** – формирование у обучающихся навыков программирования и создания алгоритмов для решения поставленных задач.

**Задачи реализации программы:**

- Изучить основы программирования на языке Python с использованием материалов образовательного ресурса Stepik.org;
- Научить применять навык алгоритмического мышления и полученные знания для решения практических задач.

- Повысить уровень толерантности к ошибкам в обучении и жизни
- Повысить уровень любознательности и самостоятельности в решении задач.

Отличительная особенность программы - проведение аудиторных занятий с использованием специально разработанных инструментов, а именно:

- онлайн-уроков, размещенных на платформе Stepik;
- проектных заданий для создания учащимися игр в среде Python;
- методологии развития soft-skills;
- сюжетной геймификации курса.

Курс предполагает смешанный формат обучения. Сочетание групповой работы с учителем в классе и индивидуальной работы в личном кабинете на онлайн-платформе позволяет ученикам выработать не только технические навыки программирования, но и навыки социального взаимодействия при работе над финальным проектом курса, а главное – научиться самостоятельно выстраивать свое профессиональное развитие.

Аттестация проводится в форме выполнения общего проекта, в котором используется пройденный материал. Контроль в указанной форме осуществляется за счёт наблюдения и саморефлексии учащихся. Тестовая и отметочная форма контроля отсутствуют.

### **Планируемые результаты освоения курса**

#### **Личностные результаты:**

- формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики;
- формирование осознанного, уважительного и доброжелательного отношения к другому человеку, его мнению; готовности и способности вести диалог с другими людьми и достигать в нём взаимопонимания;
- освоение социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах, включая взрослые и социальные сообщества; участие в общественной жизни в пределах возрастных компетенций с учётом региональных, этнокультурных, социальных и экономических особенностей;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной, творческой и других видов деятельности.

#### **Метапредметные результаты:**

По окончании курса учащийся сможет:

- Лояльно относиться к своим ошибкам;
- Анализировать свои ошибки и извлекать из них опыт;
- Понимать область своих интересов;
- Самостоятельно искать ответы, задавать вопросы;
- Презентовать свои идеи.

#### **Предметные результаты:**

По окончании курса учащийся сможет:

- Программировать в среде Python;
- Самостоятельно реализовывать проекты, связанные с разработкой игр.

**Практическим результатом** работы служит финальный проект каждого ученика: игровая программ

## Содержание курса

### **Техника безопасности и правила поведения в компьютерном классе**

Основные требования и правила поведения в компьютерном классе. Техника безопасности при работе с электрическими приборами и правила пожарной безопасности. Здоровьесберегающие технологии

### **Язык программирования Python**

История развития языков программирования. Компилируемые и интерпретируемые языки программирования, достоинства и недостатки. Установка Python 3+. Интерфейс среды программирования IDLE. Первая программа

### **Операторы ввода, вывода. Переменные**

Операторы print() и input() и их параметры. Сохранение значений в переменных. Программа "Знакомство", игра "Шутки". Программируем дружелюбный пользовательский интерфейс. Псевдографика. Осваиваем клавиатуру и особенности вывода текста на нескольких строках. Функции символа «\»

### **Типы данных, оператор присваивания**

Данные, типы данных, оператор присваивания. Функции преобразования типов. Вычисление арифметических выражений. Конкатенация строк

### **Целочисленная арифметика**

Арифметические операции. Целочисленное деление и остаток от целочисленного деления. Пишем программу, в которой Питон отгадывает задуманное число. Учащиеся придумывают и реализуют свои алгоритмы отгадывания числа

**Списки.** Списки как тип данных. Перебор элементов списка. Программа "Генеральная уборка" (простой вариант)

### **Простые условия**

Проверка условий. Сравним значения. Как и где можно записать условие. Операторы ветвления (if) и цикла (while). Игра "Гадание по ромашке", программируем двумя способами. Строковые функции upper() и lower()

### **Случайные значения. Игра "Царство драконов"**

Модули в языке Python. Модуль random. Модуль time. Игра "Царство драконов". Анализ игры. Создание блок-схемы и программы

### **Безопасность в Интернете**

Каким должен быть хороший пароль? Создаём программу для генерации хороших паролей. Повторяем всё, что изучили

### **Работа со списками**

Функция len(). Метод списка append(). Проект "Генеральная уборка". Создание списка работников в интерактивном режиме

### **Исполнитель «Черепашка»**

Модуль turtle. Знакомимся с черепашкой, среда и система команд исполнителя. Цикл for, функция range(). Черепашка рисует правильные многоугольники. Функции в Python. Обмен данными. Черепашка рисует звёзды и снежинки. Кодирование цвета. RGB-кодировка. Создание случайного цвета. Проект «Мы с черепашкой смогли...».

### **Игра "Виселица"**

Игра "Виселица". Изучение и анализ игры. Составление блок-схемы игры. Обзор функций игры. Основная программа игры. Глобальные переменные и основной цикл. Функции `getRandomWord()`, `displayBoard()`. Функции `getGuess()`, `playAgain()`. Модуль `string`. Тестирование игры. Подведение итогов. Работа со словарём. Доработка игры "Виселица". Делаем три уровня сложности. Вносим изменения в основную программу, `HANGMAN_PICS` и его обработку, в зависимости от уровня сложности. Вносим изменения в `displayBoard()`, `getRandomWord()`. Тестируем игру

### **Искусственный интеллект**

Искусственный интеллект — что это такое. Проектируем игру "Крестики-нолики". Запись сложных условий в языке Python. Делаем функции `drawBoard()`, `inputPlayerLetter()` и `whoGoesFirst()`. Стратегия игры искусственного интеллекта. Состояние игрового поля и выбор хода. Делаем функции `makeMove()`, `isBoardFull()`, `isWinner()`, `getComputerMove()`. Ссылки на список. Передача списка в функцию. Доделываем и тестируем игру "Крестики-нолики".

### **Графический пользовательский интерфейс (GUI)**

Графический пользовательский интерфейс. Модуль `tkinter`. Окна. События. Кнопки. Вспоминаем кодирование цвета и пишем программу "Радуга"

### **Криптография**

Криптография. Шифр Атбаш. Программа для шифрования и дешифрования. Шифр Цезаря. Программа для шифрования и дешифрования.

### **Таблицы**

Двумерные списки (таблицы). Упаковщик `grid()`. Ещё раз "Крестики-нолики". Создание приложения (exe-файла)

### **Игра "Сапёр"**

Анализ игры. Создание блок-схемы и обзор функций программы. Кодирование игры. Тестирование игры

### **Игра "Мемори"**

Игра "Мемори". Анализ игры. Как отключить кнопку. Как загрузить изображение. Составление блок-схемы игры. Обзор функций игры. Программируем игру "Мемори". Тестируем игру "Мемори"

### **Продвинутая графика с tkinter**

Продвинутая графика с `tkinter`. `Canvas`. Рисуем мышкой. Сдвиг объектов в `Canvas`. Как работает `coords`. Текст в `Canvas`, `create_arc`

### **Игра "Поймай шарик"**

Игра "Поймай шарик". Знакомство и обсуждение. Составление блок-схемы игры. Обзор функций игры. Программируем игру "Поймай шарик". Тестируем игру "Поймай шарик"

### **Основные формы организации занятий:**

- Практические занятия с использованием онлайн-платформы: `Stepik.org`, групповые и индивидуальные проекты и мини-игры с вариативным оформлением;
- Изучение материалов и прохождение тестовых заданий на онлайн-платформе `Stepik.org`;
- Групповые и индивидуальные задания, направленные на повторение материала, развитие `soft-skills` и рефлекссию занятий.

## Тематическое планирование курса

№	Тема	Количество часов
1	Техника безопасности и правила поведения в компьютерном классе	1
2	Язык программирования Python	1
3	Операторы ввода, вывода. Переменные	1
4	Типы данных, оператор присваивания	1
5	Целочисленная арифметика	1
6	Списки	1
7	Ветвление и цикл	2
8	Безопасность в Интернете	1
9	Случайные значения. Игра "Царство драконов"	1
10	Исполнитель «Черепашка»	4
11	Игра "Виселица"	4
12	Искусственный интеллект	4
13	Графический пользовательский интерфейс	1
14	Таблицы	2
15	Игра "Сапёр"	2
16	Игра "Мемори"	3
17	Продвинутая графика с tkinter	2
18	Игра "Поймай шарик"	2
	<b>Итого:</b>	<b>34</b>