




государственное бюджетное учреждение
Калининградской области
профессиональная образовательная организация
«Прибалтийский судостроительный техникум»

СОГЛАСОВАНО

Начальник центра
цифрового образования «IT-куб»

 А. Г. Челядинский

« 30 » июня 2020 г.

УТВЕРЖДАЮ

Директор ГБУ КО ПОО

«Прибалтийский

судостроительный техникум»

 А. В. Лукин

2020 г.



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА
«ПРОГРАММИРОВАНИЕ НА PYTHON»



2020 г.

Организация-разработчик: государственное бюджетное учреждение Калининградской области профессиональная образовательная организация «Прибалтийский судостроительный техникум» (далее – ГБУ КО ПОО «Прибалтийский судостроительный техникум»).

Составители:

Хорольская Е.В. - методист центра цифрового образования «IT-Cube» при ГБУ КО ПОО «Прибалтийский судостроительный техникум»

Программа рассмотрена на заседании методического совета ГБУ КО ПОО «Прибалтийский судостроительный техникум»

протокол № 11 от «18» 06 2020 г.

РАЗДЕЛ №1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная программа (далее – ДОП) «ПРОГРАММИРОВАНИЕ НА PYTHON» разработана в соответствии с требованиями нормативных документов:

- ФЗ РФ от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Указ Президента РФ от 7.05.2012 г. № 599 «О мерах по реализации государственной политики в области образования и науки»;
- Концепция развития дополнительного образования детей, утверждена распоряжением Правительства РФ от 4 сентября 2014 г. № 1726-р;
- Приказ Минобрнауки РФ от 29.08.2013 г. № 1008 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Примерные требования к содержанию и оформлению образовательных программ дополнительного образования детей (письмо Минобрнауки РФ от 11.12.2006 № 06-1844);
- Постановление Главного государственного санитарного врача РФ от 4.07.2014 г. № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей»
- Методические рекомендации по проектированию дополнительных общеразвивающих программ (Письмо Минобрнауки РФ от 18.11.2015).

Направленность программы: техническая.

Актуальность программы

В настоящее время мы переживаем большие изменения в развитии общества. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причем зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие обучающимися основ современного программирования и неумение применять полученные знания на практике.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. В данной программе выбран язык программирования Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирова-

ния, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Педагогическая целесообразность программы

Овладев навыками программирования на языке Python, обучающиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит обучающимся потом с легкостью освоить любой другой язык программирования.

Знания и умения, приобретенные в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

Данная программа дает возможность детям творчески мыслить, находить самостоятельные индивидуальные решения, а полученные умения и навыки применять в жизни. Развитие творческих способностей помогает также в профессиональной ориентации подростков.

Форма обучения: очная или очно-заочная с применением дистанционных образовательных технологий.

Адресат программы – дети от 12 до 16 лет. Наполняемость группы 5 – 12 человек, группы могут быть разновозрастными.

Объем и срок освоения программы, режим занятий. Программа рассчитана на 1 года обучения, занятия проходят 2 раза в неделю по 2 академических часа, всего 144 академических часа.

2. ЦЕЛИ И ЗАДАЧИ ПРОГРАММЫ

Цели программы:

- создание условий для изучения методов программирования на языке Python;
- рассмотрение различных парадигм программирования, предлагаемых этим языком (процедурная, функциональная, объектно-ориентированная);
- подготовка к использованию как языка программирования, так и методов программирования на Python в учебной и последующей профессиональной деятельности в различных предметных областях.

Задачи:

Образовательные:

- познакомить с принципами и методами фундаментального программирования;
- познакомить с принципами и методами объектно-ориентированного программирования;
- привить навыки работы в интегрированной среде разработки на языке Python;
- изучить конструкции языка программирования Python;
- привить навыки разработки эффективных алгоритмов и программ на основе

изучения языка программирования Python;

- дать представление о значении информационных технологий в развитии общества и в изменении характера труда человека;
- формировать и развивать навыки публичного выступления.

Воспитательные:

- воспитать мотивацию учащихся к изобретательству, созданию собственных программных реализаций;
- привить стремление к получению качественного законченного результата в проектной деятельности;
- привить информационную культуру: ответственное отношение к информации с учетом правовых и этических аспектов её распространения, избирательного отношения к полученной информации;
- формировать правильное восприятие системы ценностей, принципов, правил информационного общества;
- формировать потребность в самостоятельном приобретении и применении знаний, потребность к постоянному саморазвитию;
- воспитывать социально-значимые качества личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.

Развивающие:

- способствовать развитию навыков алгоритмического и логического мышления, грамотной разработки программ;
- способствовать приобретению навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;
- развивать познавательные способности ребенка, память, внимание, пространственное мышление, аккуратность и изобретательность;
- формировать творческий подход к поставленной задаче;
- развивать навыки инженерного мышления, умения работать как по предложенным инструкциям, так и находить свои собственные пути решения поставленных задач;
- развивать навыки эффективной деятельности в проекте;
- развивать стрессоустойчивость;
- развивать способности к самоанализу, самопознанию;
- формировать навыки рефлексивной деятельности.

3. СОДЕРЖАНИЕ ПРОГРАММЫ

3.1. Учебный план

№	Наименование тем раздела	Количество часов		
		Всего	Теория	Практика
	Модуль 1. Введение в программирование.	30	14	16
1	Тема 1.1. Знакомство с системой Яндекс. Контекст. Понятия кода, интерпретатора, программы. Интегрированные среды, исполнение кода. Простейшие программы с выводом на экран.	3	2	1
2.	Тема 1.2. Условный оператор. Переменные и арифметика.	3	2	1
3	Тема 1.3. Знакомство с циклом while.	4	2	2
4	Тема 1.4. Отладчик.	4	2	2
5	Тема 1.5. Знакомство с циклом for.	4	2	2
6	Тема 1.6. Погружение в условия. True, False, Break, Continue.	4	2	2
7	Тема 1.7. Знакомство со списками.	4	2	2
8	Тема 1.8. Решение задач по теме модуля 1.	4		4
	Модуль 2. Базовые конструкции в Python.	42	20	22
1	Тема 2.1. Дополнительные возможности цикла for. Кортежи. Сортировки.	4	2	2
2	Тема 2.2. Знакомство со срезами и диапазонами. Равенство и совпадение объектов.	4	2	2
3	Тема 2.3. Списочные выражения. Методы split и join.	4	2	2
4	Тема 2.4. Другие методы списков и строк.	4	2	2
5	Тема 2.5. Функции.	6	4	2
6	Тема 2.6. Области видимости переменных.	6	4	2
7	Тема 2.7. Индивидуальные задания по темам 1 и 2 модулей.	2		2
8	Тема 2.8. Разбор выполнения заданий.	2		2
9	Тема 2.9. Введение в компьютерную графику.	6	4	2

	Модуль TKInter.			
10	Тема 2.10. Решение задач по теме модуля.	4		4
	Модуль 3. Проекты GUI, Игры и API.	72	18	54
1	Тема 3.1. Изучение виджетов.	10	6	4
2	Тема 3.2. Решение задач на тему «GUI».	4		4
3	Тема 3.3. Самостоятельная работа.	2		2
4	Тема 3.4. Работа над проектом по теме «GUI» и его защита.	8		8
5	Тема 3.5. Изучение игровых технологий и средств языка Python для их реализации.	14	6	8
6	Тема 3.6. Самостоятельная работа.	2		2
7	Тема 3.7. Работа над проектом по теме «Игры» и его защита.	8		8
8	Тема 3.8. Изучение некоторых API и работа с ними с помощью языка Python.	14	6	8
9	Тема 3.9. Самостоятельная работа.	2		2
10	Тема 3.10. Работа над проектом по теме «API» и его защита.	8		8
	ИТОГО	144	52	92

3.2.1. Содержание учебно-тематического плана

Модуль 1. Введение в программирование.

Знакомство с виртуальной средой взаимодействия: регистрация, организация личного кабинета, поиск и выкладывание материалов. Знакомство с системой автоматизированной проверки задач.

Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки, интерпретатор, код программы и редактор кода.

Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

На этом этапе обучающиеся разрабатывают первые алгоритмы и программы, а

также анализируют, на какие функциональные блоки может быть разбита программа, и определяют работоспособность разработанной программы.

В результате освоения модуля 1 обучающиеся

должны знать:

- базовые конструкции языка программирования Python (операции присваивания, ветвления, цикла, ввод\вывод, запись констант и выражений);
- основные методы сортировки списков;
- алгоритмы обработки списков (поиск элемента с заданными свойствами, удаление элементов, вставка элементов, замена элементов).

должны уметь:

- устанавливать необходимый инструментарий для разработки приложений;
- составлять программный код с использованием разных типов данных и операций над ними;
- составлять, отлаживать и исполнять на персональном компьютере программы решения учебных задач по программированию;
- применять условные конструкции;
- решать задачи с использованием разных видов циклов;
- составлять программы обработки списков;
- искать информацию в свободных источниках и структурировать ее;
- объективно оценивать результаты своей работы.

Тема 1.1. Знакомство с системой *Яндекс.Контекст*.

Теория. Понятия кода, интерпретатора, программы. Интегрированные среды, исполнение кода.

Практика. Простейшие программы с выводом на экран.

Тема 1.2. Условный оператор. Переменные и арифметика.

Теория. Алфавит и словарь языка Python. Объявление переменных. Типы данных. Арифметические операции. Операции с присваиванием. Линейные программы. Условный оператор.

Практика. Обзор типов данных. Вычисление математических выражений с помощью стандартных арифметических функций. Решение задач по теме.

Тема 1.3. Знакомство с циклом `while`.

Теория. Определение цикла. Оператор цикла с предусловием.

Практика. Решение задач по теме «Цикл `while`».

Тема 1.4. Отладчик.

Теория. Определение. Установка и запуск отладчика программного кода.

Практика. Решение задач.

Форма подведения итогов: самостоятельная работа на решение задач по теме 1.4.

Тема 1.5. Знакомство с циклом `for`.

Теория. Оператор цикла с постусловием. Оператор с известным числом повторов.

Практика. Решение задач по теме «Цикл `for`».

Тема 1.6. Погружение в условия. `True`, `False`, `Break`, `Continue`.

Теория. Принцип работы операторов `True`, `False`, `Break`, `Continue`.

Практика. Решение задач на тему «Погружение в условия».

Тема 1.7. Знакомство со списками.

Теория. Определение понятия «списки». Объявление списка. Действия над списками, над элементами списка.

Практика. Решение задач по теме «Списки»

Форма подведения итогов: самостоятельная работа по теме модуля.

Модуль 2. Базовые конструкции языка Python.

Понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python.

Устройство циклов for. Основные управляющие конструкции циклического алгоритма в Python. Простейшие циклы и циклы с переменными.

Работа со списками, строками, множествами и кортежами в Python. Понятие итератора.

Понятие подпрограммы, процедуры, функции. Функции в Python. Решение задач.

В результате освоения модуля 2 обучающиеся

должны знать:

- базовые конструкции языка программирования Python (операции присваивания, ветвления, цикла, ввод\вывод, запись констант и выражений);
- основные методы сортировки списков;
- алгоритмы обработки списков (поиск элемента с заданными свойствами, удаление элементов, вставка элементов, замена элементов);
- способы организации процедур и функций.

должны уметь:

- записывать конструкции языка программирования Python;
- тестировать и осуществлять отладку программ;
- объективно оценивать результаты своей работы.

Тема 2.1. Повторение. Дополнительные возможности цикла for.

Кортежи. Сортировки.

Теория. Структура программы. Стандартные функции и арифметические выражения. Ветвления. Логические операции. Оператор цикла с известным количеством повторов, постусловием, предусловием. Оператор безусловного перехода.

Практика. Решение задач на тему «Цикл for».

Тема 2.2. Знакомство со срезами и диапазонами.

Теория. Понятие срезов и диапазонов. Равенство и совпадение объектов.

Практика. Решение задач по теме «Срезы и диапазоны»

Тема 2.3. Списочные выражения. Методы split и join.

Теория. Работа со списками. Применение методов split и join.

Практика. Решение задач на тему «Методы split и join».

Тема 2.4. Другие методы списков и строк.

Теория. Генераторы списков. Методы find и rfind. Метод replace и count.

Практика. Решение задач на методы списков и строк.

Тема 2.5. Знакомство с функциями.

Теория. Определение функции. Правила составления и вызова функций в программном коде. Локальные и глобальные переменные.

Практика. Решение задач на тему «Функции».

Тема 2.6. Области видимости переменных.

Теория. Локальная область видимости. Глобальная область видимости. Нелокальная область видимости

Практика. Решение задач на тему «Области видимости переменных».

Тема 2.7. Введение в компьютерную графику. Модуль TKInter.

Теория. Компьютерная графика и её виды. Кроссплатформенная библиотека для разработки графического интерфейса TKInter.

Практика. Решение задач на тему «Модуль TKInter».

Форма подведения итогов: выполнение индивидуальных заданий по теме модуля.

Модуль 3. Проекты GUI, Игры и API

Модуль GUI открывает группу проектных тем курса. В теме «GUI» обучающиеся рассматривают принципы создания пользовательских интерфейсов на языке Python с использованием соответствующих библиотек.

Тема «Игры» предлагает к изучению различные виды компьютерных игр, технологии и алгоритмы для их проектирования. Тема «API» раскрывает способы интеграции различных сетевых приложений и сервисов.

Каждая тема содержит мини-проекты и итоговый проект, который обучающиеся по желанию защищают на итоговом занятии.

В результате освоения модуля 3 обучающиеся должны знать:

- основные понятия объектно-ориентированного проектирования;
- способы интеграции различных сетевых приложений и сервисов;
- принципы создания пользовательских интерфейсов на языке Python с использованием соответствующих библиотек.

должны уметь:

- работать с информацией: находить, оценивать и использовать информацию из различных источников;
- разрабатывать эффективные алгоритмы и программ на основе изучения языка программирования Python;
- использовать специальные средств и библиотеки языка Python;
- самостоятельно изучать новые технологии;
- работать в команде и использовать соответствующие технологии для организации командной работы;
- генерировать идеи;
- объективно оценивать результаты своей работы.

Тема 3.1. Изучение виджетов.

Теория. Обзор библиотек для создания графического интерфейса пользователя. Перетаскивание (drag and drop). Диалоговые окна. События и сигналы. Управление макетом. Меню и тулбары.

Практика. Создание собственного виджета в PyQt5.

Тема 3.2. Решения задач на тему «GUI».

Практика. Решения задач на тему 3.2.

Форма подведения итогов. Самостоятельная работа.

Тема 3.3. Работа над проектом по теме «GUI» и его защита.

Практика. Организация работы над проектом. Постановка проблемной ситу-

ации, формулировка цели и задач. Работа над проектом. Подготовка к защите.

Форма подведения итогов. Публичная защита проекта.

Тема 3.4. Изучение игровых технологий и средств языка Python для их реализации.

Теория. Классические игры на клетчатом поле. Изображения. Спрайты. Столкновения и другие взаимодействия. Украшения игры.

Практика. Решения задач на тему «Игры».

Форма подведения итогов. Самостоятельная работа.

Тема 3.5. Работа над проектом по теме «Игры» и его защита.

Практика. Организация работы над проектом. Постановка проблемной ситуации, формулировка цели и задач. Работа над проектом. Подготовка к защите.

Форма подведения итогов. Публичная защита проекта.

Тема 3.6. Изучение некоторых API и работу с ними с помощью языка Python.

Теория. Знакомство с API. Работа с протоколом HTTP. Введение в Telegram Bot API.

Практика. Решение задач на API Яндекс.Карт. Создание сценариев диалогов в Telegram Bot API.

Форма подведения итогов. Самостоятельная работа.

Тема 3.7. Работа над проектом по теме «API» и его защита.

Практика. Организация работы над проектом. Постановка проблемной ситуации, формулировка цели и задач. Работа над проектом. Подготовка к защите.

Форма подведения итогов. Публичная защита проекта.

4. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

4.1. Планируемые результаты:

Основным результатом обучения является достижение высокой информационно-коммуникационной компетентности обучающегося в области программирования на языке Python.

В результате освоения программы обучающийся должен приобрести следующие знания, умения и навыки:

знать:

- правила работы с компьютером и технику безопасности;
- основные предметные понятия («информация», «алгоритм», «исполнитель», «модель») и их свойства;
- основы современных языков программирования;
- особенности работы с интегрированной средой разработки;
- базовые и сложные конструкции, способы организации процедур и функций в языке программирования Python;
- технику ведения проектной деятельности и принципов тайм-менеджмента.

уметь:

- работать с информацией: находить с применением правил поиска в компьютерных сетях, оценивать и использовать информацию из различных источников при выполнении заданий и проектов по различным темам;
- определять результат выполнения алгоритма при заданных исходных данных,

- узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе программы анализа данных, читать и понимать программы, написанные на языке программирования высокого уровня Python;
- выполнять пошагово алгоритмы управления исполнителями и анализа числовых и текстовых данных;
 - разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
 - самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
 - самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата;
 - критически оценивать правильность решения учебно-исследовательской задачи;
 - корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
 - организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности

обладать навыками:

- исследовательской, проектной и социальной деятельности, строить логическое доказательство;
- использования, создания и преобразования различных символьных записей, схем и моделей для решения познавательных и учебных задач в различных предметных областях, исследовательской и проектной деятельности;
- разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- использования специальных средств и библиотек языка Python
- работы в интегрированной среде разработки на языке программирования Python;
- алгоритмического и логического мышления, грамотной разработки программ;
- проектирования, разработки, документирования и представления собственных проектов;
- самообразования - периодической оценкой своих успехов и собственной работы самими обучающимися.

4.2. Формы оценки уровня достижений обучающегося

Для контроля и самоконтроля за эффективностью обучения применяются методы:

- предварительные (диагностика, наблюдение);
- текущие (наблюдение);
- тематические (контрольные вопросы, промежуточные задания);
- итоговые (проект, итоговое тестирование).

Динамика развития познавательных способностей оценивается по качеству выполнения практикумов и мини- проектов. Сопоставляя успешность реализации предыдущего проекта с текущим, отслеживается динамика роста познавательных способностей обучающихся.

Формы фиксации образовательных результатов

Для фиксации образовательных результатов в рамках курса используются:

- электронный журнал;
- портфолио работ учащихся;
- отзывы обучающихся по итогам занятий и итогам обучения.

Формы предъявления и демонстрации образовательных результатов:

- итоговое занятие с представлением итоговых проектов, итоговое тестирование.

Формы подведения итогов реализации программы

- педагогическое наблюдение;
- педагогический анализ выполнения обучающимися учебных заданий;
- защита проектов;
- активность обучающихся на занятиях и т.п.

РАЗДЕЛ №2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Материально-техническое обеспечение программы

Для реализации данной образовательной программы необходимо определённое методическое обеспечение:

- компьютерный класс.
- программное обеспечение на ПК.
- доступ в сеть Интернет.
- интерактивная панель.

Информационно-методическое обеспечение программы

Тема модуля	Форма занятий	Приёмы и методы организации образовательного процесса	Дидактический материал. Электронные источники	Техническое оснащение и расходный материал	Форма подведения итогов
Модуль 1. Введение в программирование	Комбинированная	Метод проектов. Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.	1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru); 2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (http://anytask.org); 3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com); 4. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru); 6. Официальная документация языка Python (http://docs.python.org).	<ul style="list-style-type: none"> ● Ноутбуки с мышкой и доступом к сети Интернет. ● Презентационное оборудование. ● Принтер. ● Маркерная доска. 	Выполнение индивидуальных заданий.
Модуль 2. Базовые конструкции в Python	Комбинированная	Метод проектов. Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.	1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru); 2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (http://anytask.org); 3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com); 4. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru); 6. Официальная документация языка Python (http://docs.python.org).	<ul style="list-style-type: none"> ● Ноутбуки с мышкой и доступом к сети Интернет. ● Презентационное оборудование. 	Выполнение индивидуальных заданий.

		блемно-поисковый.	<p>батывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru);</p> <p>2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (http://anytask.org);</p> <p>3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com);</p> <p>4. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru);</p> <p>6. Официальная документация языка Python (http://docs.python.org).</p>	<p>тационное оборудование.</p> <ul style="list-style-type: none"> ● Принтер. ● Маркерная доска. 	даний.
Модуль 3. Проекты GUI, Игры и API	Комбинированная	<p>Метод проектов. Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.</p>	<p>1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru);</p> <p>2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (http://anytask.org);</p> <p>3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com);</p> <p>4. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru);</p> <p>6. Официальная документация языка Python (http://docs.python.org).</p>	<ul style="list-style-type: none"> ● Ноутбук с мышкой и доступом к сети Интернет. ● Презентационное оборудование. ● Принтер. ● Маркерная доска. 	Защита проекта.

Список литературы

1. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.
2. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И. Г. Семкина и Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.
3. Материалы и презентации к урокам в LMS Яндекс.Лицея.
4. Сайт pythonworld.ru — «Python 3 для начинающих».
5. Сайт pythontutor.ru — «Питонтьютор».
6. <https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwBI> — Лекции А.В. Умнова, прочитанные в Школе Анализа Данных Яндекса.