



МИНОБРНАУКИ РОССИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
МОРДОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. Н.П. ОГАРЁВА»**

(ФГБОУ ВО «МГУ им. Н.П. Огарёва»)

УТВЕРЖДЕНО

решением учёного совета
ФГБОУ ВО «МГУ им. Н.П. Огарёва»
(протокол от «27» июня 2023 г. № 12)

Председатель ученого совета
Ректор _____ Д.Е. Глушко



**Дополнительная общеобразовательная
общеразвивающая программа**

***Лицей Академии Яндекса. 1-й год.
«Основы программирования на языке Python»***

Форма обучения – очная
Объем программы: 144 часов
Срок обучения: 9 месяцев

Саранск 2023

Разработчик(и)
ДООП

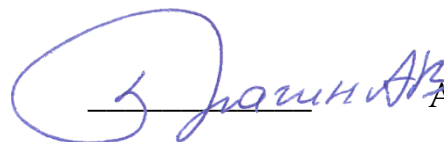
педагог



А.А. Попов

Согласовано:

Директор ДНК им.
академика Е.М.
Дианова



А.В. Брагин

«11» мая 2023 г.

Директор института
непрерывного
обучения и
корпоративного
образования



Н.В. Жадунова
«15» мая 2023 г.

Структура программы

1. Пояснительная записка.	3
1.1 Направленность программы.	3
1.2 Актуальность программы.	3
1.3 Цель программы.	4
1.4 Задачи программы.	4
1.5 Отличительные особенности программы, новизна.	4
1.6 Нормативные правовые акты, на которых базируется разработка программы	5
1.7 Адресат программы.	6
1.8 Объем программы	6
1.9 Срок освоения:	6
1.10 Форма обучения:	6
1.11 Особенности организации образовательного процесса.	6
1.12 Организационные формы обучения:	7
1.13 Режим занятий:	7
1.14 Планируемые результаты освоения программы	7
1.15 Документ об обучении, выдаваемый по окончании обучения, и условия его получения обучающимся.	9
2. Учебный план и учебно-тематический план	10
2.1 Учебный план	10
2.2 Учебно-тематический план	11

3. Рабочие программы учебных предметов, курсов, дисциплин (модулей)	15
3.1 Программа модуля 1 «Введение в программирование»	15
3.2 Программа модуля 2 «Базовые конструкции в Python»	17
3.3 Программа модуля 3 «Решение прикладных задач в Python»	20
4. Календарный учебный график дополнительной общеобразовательной общеразвивающей программы «Основы программирования на языке Python (углубленный уровень)»	27
5. Организационно-педагогические условия реализации программы	30
5.1 Кадровое обеспечение	30
5.2 Учебно-методическое и информационное обеспечение	31
5.3 Материально-техническое обеспечение	32
6. Нормативно-методическое обеспечение системы оценки качества освоения обучающимися дополнительной общеобразовательной общеразвивающей программы	33
6.1 Формы аттестации	33
6.2 Оценочные материалы	33
6.2.1 Комплект оценочных средств для проведения промежуточной аттестации	33
6.2.2 Комплект оценочных средств для проведения итоговой аттестации	34
7. Сведения об обновлении программы	38

1. Пояснительная записка.

1.1 Направленность программы.

Техническая.

1.2 Актуальность программы.

В настоящее время мы переживаем большие изменения в развитии общества. 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.

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 учащимися основ современного программирования и неумение применять полученные знания на практике.

Опираясь на уникальный опыт преподавания программирования в АНО ДПО «Образовательные технологии Яндекса» и на факультете компьютерных наук НИУ ВШЭ, была подготовлена данная программа. В ней большое внимание уделяется практической работе на компьютере и самостоятельному написанию кода.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык 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 программирования, 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.

Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые 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.

1.3 Цель программы.

Целью курса является 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.

1.4 Задачи программы.

- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- знакомство с принципами и методами функционального программирования;
- знакомство с принципами и методами объектно-ориентированного программирования;
- приобретение навыков работы в интегрированной среде разработки на языке Python;
- изучение конструкций языка программирования Python;
- знакомство с основными структурами данных и типовыми методами обработки этих структур;
- приобретение навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;
- развитие у обучающихся интереса к программированию;
- формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- воспитание упорства в достижении результата;
- расширение кругозора обучающихся в области программирования.

1.5 Отличительные особенности программы, новизна.

- обеспечивает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном уровне;
- имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика;
- охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний;
- ориентирована на существующий парк вычислительной техники и дополнительные ограничения;

– допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня учащихся (как группового, так и индивидуального);

– предусматривает возможность индивидуальной работы с учащимися.

Практическая значимость курса заключается в том, что он способствует более успешному овладению знаниями и умениями по направлению «Программирование» через развитие самостоятельности обучающихся и оптимизацию средств и методов обучения.

Элементы программы курса могут быть рекомендованы для использования учителями информатики при проведении лабораторно-практических и практических занятий.

1.6 Нормативные правовые акты, на которых базируется разработка программы

– Федеральный закон от 29 декабря 2012 года №273-ФЗ «Об образовании в Российской Федерации»;

– – Постановление Правительства Республики Мордовия от 22 августа 2019 года № 352 «Об утверждении Порядка предоставления из республиканского бюджета Республики Мордовия гранта в форме субсидии некоммерческим организациям на обеспечение расходов по содержанию центров, реализующих дополнительные общеобразовательные программы, в организациях, осуществляющих образовательную деятельность по образовательным программам высшего образования, в том числе участвующих в создании научных и научно-образовательных центров мирового уровня или обеспечивающих деятельность центров компетенций Национальной технологической инициативы»;

– Распоряжение Правительства РФ от 31 марта 2022 г. N 678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 г. и плана мероприятий по ее реализации»;

– Приказ Министерства просвещения Российской Федерации от 27.07.2022 года № 629 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

– Приказ Министерства образования и науки РФ от 23.08.2017 года № 816 «Об утверждении Порядка применения организациями, осуществляющими образовательную деятельность, электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»;

– Приказ Министерства науки и высшего образования РФ и Министерства просвещения РФ от 5 августа 2020 г. N 882/391 «Об организации и осуществлении образовательной деятельности при сетевой форме реализации образовательных программ»;

– Приказ Министерства труда и социальной защиты Российской Федерации от 22 сентября 2021 г. N 652н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»;

– Приказ Министерства образования Республики Мордовия от 4 марта 2019 года № 211 «Об утверждении Правил персонифицированного финансирования дополнительного образования детей»;

– Санитарные правила СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», утвержденные постановлением Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 г. № 28 (зарегистрировано Министерством юстиции Российской Федерации 18 декабря 2020 г., регистрационный N 61573), действующие до 1 января 2027 года.

– Устав Университета и другие локальные нормативные акты Университета.

1.7 Адресат программы.

Дополнительная общеобразовательная (общеразвивающая) программа «Основы программирования на языке Python (углубленный уровень)» ориентирована на работу с учениками 8-го и 9-го классов школ разного уровня подготовки и с разной степенью мотивации.

1.8 Объем программы

Продолжительность реализации всей программы -144 часа.

1.9 Срок освоения:

Срок реализации программы - 7 месяцев.

1.10 Форма обучения:

Очная.

1.11 Особенности организации образовательного процесса.

Очная программа с применением ДОТ.

Основной тип занятий — комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок уроков (модуль). Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится учителем самостоятельно, но с учётом рекомендованного календарно-тематического плана. С учётом регулярного повторения ранее изученных тем темп изучения отдельных разделов блока определяется субъективными и объективными факторами.

Каждая тема курса начинается с постановки задачи — характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью учитель проводит демонстрацию презентации или

показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в классе. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом ученики не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

1.12 Организационные формы обучения:

В процессе реализации программы используются формы: групповая, индивидуальная. Большинство занятий проводится в групповой форме. Группы, в которых проходят занятия могут быть разновозрастными.

1.13 Режим занятий:

Занятия проводятся 2 раза в неделю по 2 часа (продолжительность учебного часа 45 минут).

1.14 Планируемые результаты освоения программы

Личностные результаты:

- 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, учебно-исследовательской 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 результаты:

- 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);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

Предметные результаты:

- 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;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;

– умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;

– навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;

– умение 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;

– формирование умения соблюдать нормы информационной этики и права.

1.15 Документ об обучении, выдаваемый по окончании обучения, и условия его получения обучающимся.

Документ об обучении – сертификат установленного образца (получают лица, освоившие программу в полном объёме и прошедшие итоговую аттестацию).

2. Учебный план и учебно-тематический план

2.1 Учебный план

дополнительной общеобразовательной общеразвивающей программы
«Основы программирования на языке Python (углубленный уровень)»

№	Наименование разделов/модулей	Всего часов	В том числе			Форма контроля
			Лекции	Практические занятия	Самостоятельная работа	
	Модуль 1. Введение в программирование.	30	10	12	8	Наблюдение, анализ работ, устный опрос
	Модуль 2. Базовые конструкции в Python.	38	13	17	8	Наблюдение, анализ работ, устный опрос, контрольная работа
	Модуль 2. Решение прикладных задач в Python.	76	29	35	12	Наблюдение, анализ работ, устный опрос, контрольная работа
	Итоговая аттестация					
	ИТОГО:	144	144	63	28	

2.2 Учебно-тематический план

дополнительной общеобразовательной общеразвивающей программы «Основы программирования на языке Python (углубленный уровень)»

№ п/п	Наименования Разделов/модулей и дисциплин/тем	Всего часов	В том числе:			Форма контроля
			лекции	практич еские занятия	самост оятельн ая работа	
Модуль 1. <u>Введение в программирование</u> (наименование модуля)						
Всего: 30 часов (из них: 10 часов – лекционных, 12 часов – практических, 8 часов – самостоятельная работа)						
1.1	Знакомство со средой	3	1	1	1	Наблюдение, опрос детей, анализ работ
1.2	Условный оператор	3	1	1	1	Наблюдение, опрос детей, анализ работ
1.3	Простые встроенные функции	3	1	1	1	Наблюдение, опрос детей, анализ работ
1.4	Знакомство с циклом while	3	1	1	1	Наблюдение, опрос детей, анализ работ
1.5	Отладчик	3	1	1	1	Наблюдение, опрос детей, анализ работ
1.6	Знакомство с циклом for	3	1	1	1	Наблюдение, опрос детей, анализ работ
1.7	true и False, break и continue	3	1	1	1	Наблюдение, опрос детей, анализ работ
1.8	Вложенные циклы	3	1	1	1	Наблюдение, опрос детей, анализ работ
1.9	Резерв (дорешка)	2	1	1	0	Наблюдение, опрос детей, анализ работ
1.10	Резерв (дорешка)	2	1	1	0	Наблюдение, опрос детей, анализ работ
1.11	Резерв (пробная СР/разбор пробной СР)	2	0	2	0	Наблюдение, опрос детей, анализ работ

Модуль 2. <u>Базовые конструкции в Python</u> (наименование модуля)						
Всего: 38 часов (из них: 13 часов – лекционных, 17 часов – практических, 8 часов – самостоятельная работа)						
2.1	Множества	3	1	1	1	Наблюдение, опрос детей, анализ работ
2.2	Строки. Индексация	3	1	1	1	Наблюдение, опрос детей, анализ работ
2.3	Строки. Срезы	3	1	1	1	Наблюдение, опрос детей, анализ работ
2.4	Знакомство со списками	2	1	1	0	Наблюдение, опрос детей, анализ работ
2.5	Кортежи. Преобразование коллекций	3	1	1	1	Наблюдение, опрос детей, анализ работ
2.6	Методы split и join. Списочные выражения	3	1	1	1	Наблюдение, опрос детей, анализ работ
2.7	Методы списков и строк	3	1	1	1	Наблюдение, опрос детей, анализ работ
2.8	Резерв (подготовка к СР/повторение)	2	1	1	0	Наблюдение, опрос детей, анализ работ
2.9	Самостоятельная работа №1	2	0	2	0	Наблюдение, опрос детей, анализ работ
2.10	Вложенные списки	3	1	1	1	Наблюдение, опрос детей, анализ работ
2.11	Знакомство со словарями	3	1	1	1	Наблюдение, опрос детей, анализ работ
2.12	Резерв (подготовка к КР/повторение)	2	1	1	0	Наблюдение, опрос детей, анализ работ
2.13	Контрольная работа №1	2	0	2	0	Наблюдение, опрос детей, анализ работ
2.14	Резерв (дорешка/tkinter/разбор КР)	2	1	1	0	Наблюдение, опрос детей, анализ работ
2.15	Резерв	2	1	1	0	Наблюдение, опрос детей, анализ работ
Модуль 3. <u>Решение прикладных задач в Python</u> (наименование модуля)						
Всего: 76 часов (из них: 29 часов – лекционных, 35 часов – практических, 12 часов – самостоятельная работа)						

3.1.	Функции	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.2.	Функции. Возвращение значений из функции	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.3.	Области видимости переменных	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.4.	Функции: передача параметров	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.5	Функции с переменным числом аргументов	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.6	Функции как объект. Лямбда-функции	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.7	Обработка коллекций. Поточковый ввод sys.stdin	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.8	Резерв (дорежка/итераторы и коллекции)	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.9	Резерв (дорежка/переопределение функции и декораторы)	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.10	Самостоятельная работа №2	2	0	2	0	Наблюдение, опрос детей, анализ работ
3.11	Резерв (Разбор СР/ДОП - Рекурсия)	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.12	Библиотеки Python. Часть №1 (Модули стандартной библиотеки)	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.13	Библиотеки Python. Часть №2 (графика)	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.14	Библиотеки Python. Часть №3 (графика + звук)	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.15	Библиотеки Python. Часть №4 (морфология)	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.16	Библиотеки Python. Часть №5 (документы)	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.17	Резерв (подготовка к СР/ДОП - NumPy)	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.18	Самостоятельная работа №3	2	0	2	0	Наблюдение, опрос детей, анализ работ

3.19	Введение в ООП	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.20	Введение в ООП. Полиморфизм	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.21	Резерв	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.22	Резерв	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.23	ООП. Определение операторов.	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.24	ООП. Наследование	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.25	ООП. Наследование. Часть №2	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.26	Проектирование и разработка классов. Часть №1	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.27	Проектирование и разработка классов. Часть №2	3	1	1	1	Наблюдение, опрос детей, анализ работ
3.28	Резерв (подготовка к КР)	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.29	Контрольная работа №2	2	0	2	0	Наблюдение, опрос детей, анализ работ
3.30	Резерв (дорешка/разбор КР)	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.31	Резерв	2	1	1	0	Наблюдение, опрос детей, анализ работ
3.32	Резерв	2	1	1	0	Наблюдение, опрос детей, анализ работ
	Итоговая аттестация	в соответствии с нормами времени				
	ИТОГО:	144	52	64	28	

3. Рабочие программы учебных предметов, курсов, дисциплин (модулей)

3.1 Программа модуля 1 «Введение в программирование»

Образовательная задача модуля:

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 течение модуля ученики решают большое количество задач: от самых простых до сложных.

Объем дисциплины (модуля): 30 часов,

в т.ч. лекций – 10 часов;

практических – 12 часов;

самостоятельной работы 8 – часов.

Содержание модуля:

Тема 1.1 Знакомство со средой (лекций 1 час, практических 1 час)

Лекция: Введение в язык Python и знакомство с теми техническими средствами, которые понадобятся для обучения.

Написание программ, которые умеют что-то выводить на экран и считывать информацию с клавиатуры. Знакомство с переменными.

Практическое занятие: Решение задач по теме урока.

Тема 1.2 Условный оператор (лекций 1 час, практических 1 час)

Лекция: Знакомство с одной из фундаментально важных тем в программировании — условным оператором. Он позволяет организовать ветвление в вашей программе (выполнение одной ветки кода в зависимости от условия).

Практическое занятие: Решение задач по теме урока.

Тема 1.3 Простые встроенные функции (лекций 1 час, практических 1 час)

Лекция: Знакомство с типами данных, научимся работать с числами и узнаем о простейших функциях.

Практическое занятие: Решение задач по теме урока.

Тема 1.4 Знакомство с циклом while (лекций 1 час, практических 1 час)

Лекция: Знакомство с оператором цикла `while`. Цикл позволяет организовать многократное повторение одних и тех же действий. Кроме того, акцент делается на том, что в одной и той же строчке программы на разных итерациях цикла переменные могут иметь разное значение.

Практическое занятие: Решение задач по теме урока.

Тема 1.5 Отладчик (лекций 1 час, практических 1 час)

Лекция: Рассмотрим работу с отладчиком на примере программы, которую нужно исправить. Затем даются группы задач, алгоритмы решения которых требуют рассмотрения многих случаев. В этом уроке нет тестов к задачам точно так же, как на контрольных и самостоятельных работах. Учащимся придется самим выяснять, на каких входных данных не работает программа.

Практическое занятие: Решение задач по теме урока.

Тема 1.6 Знакомство с циклом `for` (лекций 1 час, практических 1 час)

Лекция: Рассматриваются именованные аргументы функции `print`, специальные символы в строке и конструкция `for ... in range(...)`.

Практическое занятие: Решение задач по теме урока.

Тема 1.7 `true` и `False`, `break` и `continue` (лекций 1 час, практических 1 час)

Лекция: Рассматриваются условия выхода из циклов. Рассматривается булев тип, даются задачи на использование флагов. Затем рассматриваются операторы `break` и `continue`, позволяющие в некоторых случаях избавиться от флагов.

Практическое занятие: Решение задач по теме урока.

Тема 1.8 Вложенные циклы (лекций 1 час, практических 1 час)

Лекция: Рассматриваются вложенные циклы, позволяющие запустить цикл внутри циклического оператора. Приводится несколько примеров вложенности разных циклов, а также применение операторов `break` и `continue` со вложенными циклами.

Практическое занятие: Решение задач по теме урока.

Тема 1.9 Резерв (дорешка) (лекций 1 час, практических 1 час)

Лекция: Повторение материала пройденных тем, разбор задач.

Практическое занятие: Решение не решенных задач по пройденным темам.

Тема 1.10 Резерв (дорешка) (лекций 1 час, практических 1 час)

Лекция: Повторение материала пройденных тем, разбор задач.

Практическое занятие: Решение не решенных задач по пройденным темам.

Тема 1.11 Резерв (пробная СР/разбор пробной СР) (лекций 1 час, практических 1 час)

Лекция: Разбор задач пробной СР

Практическое занятие: Решение задач пробной СР.

Задания для самостоятельной работы по модулю: Решение задач по пройденным темам.

Перечень ресурсов Учебные материалы и задания по темам на платформе LMS Лицея Академии Яндекса <https://lyceum.yandex.ru/>.

3.2 Программа модуля 2 «Базовые конструкции в Python»

Образовательная задача модуля:

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.

Объем дисциплины (модуля): 38 часов,

в т.ч. лекций – 13 часов;

практических – 17 часов;

самостоятельной работы – 8 часов.

Содержание модуля

Тема 2.1 Множества (лекций 1 час, практических 1 час)

Лекция: Знакомство с множествами в Python. Этот тип данных аналогичен математическим множествам, он поддерживает быстрые операции проверки наличия элемента в множестве, добавления и удаления элементов, операции объединения, пересечения и вычитания множеств.

Практическое занятие: Решение задач по теме урока.

Тема 2.2 Строки. Индексация (лекций 1 час, практических 1 час)

Лекция: Более подробное рассмотрение строкового типа данных. Изучим как не только считывать строку, но и работать с ней, в том числе делать посимвольный перебор.

Практическое занятие: Решение задач по теме урока.

Тема 2.3 Строки. Срезы (лекций 1 час, практических 1 час)

Лекция: Продолжение отработки навыков работы со строкой и знакомство с новым методом извлечения подстроки — срез.

Практическое занятие: Решение задач по теме урока.

Тема 2.4 Знакомство со списками (лекций 1 час, практических 1 час)

Лекция: Знакомство с новым типом данных — список (list), обращение к элементам списка по индексу (аналогично строкам, но с возможностью изменения

элементов списка), метод `append` и вопросы перебора элементов списка и срезов списка.

Практическое занятие: Решение задач по теме урока.

Тема 2.5 Кортежи. Преобразование коллекций (лекций 1 час, практических 1 час)

Лекция: Знакомство с новым контейнером — кортеж (`tuple`). Более подробно рассматривается операция присваивания кортежей, знакомая нам по конструкции `a, b = b, a`, и применение этой операции в реализации классического алгоритма — сортировки пузырьком. Рассматриваются и вопросы преобразования одной коллекции в другую.

Практическое занятие: Решение задач по теме урока.

Тема 2.6 Методы `split` и `join`. Списочные выражения (лекций 1 час, практических 1 час)

Лекция: Знакомство с двумя очень популярными методами строк — `split` и `join`. Во второй части рассматриваются списочные выражения (`python list comprehensions [... for ... in ...]`), которые в сочетании с методом `split` позволяют, например, считывать из стандартного потока ввода несколько значений одной строкой.

Практическое занятие: Решение задач по теме урока.

Тема 2.7 Методы списков и строк (лекций 1 час, практических 1 час)

Лекция: В материалах урока приводятся таблицы с почти полным перечнем методов списков и строк, которые можно использовать как справочный материал. Рассматривается неявное приведение объектов к булеву типу. Приводятся примеры цепочек вызова методов. Появление метода `pop` позволяет познакомиться с понятием стека.

Этот урок отличается от прочих: большую часть его материала не нужно запоминать, но можно использовать как справочный материал. С этой же целью вводятся функции `dir` и `help`.

Практическое занятие: Решение задач по теме урока.

Тема 2.8 Резерв (подготовка к СР/повторение) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала. Разбор с учениками задач из классных и домашних работ по темам из будущей СР.

Практическое занятие: Решение задач по пройденным темам.

Тема 2.9 Самостоятельная работа №1 (лекций 0 часов, практических 2 часа)

Практическое занятие: Решение задач из самостоятельной работы, которая состоит из трёх задач, которые можно решать в любом порядке.

Тема 2.10 Вложенные списки (лекций 1 час, практических 1 час)

Лекция: Так как элементами списка могут быть любые объекты — числа, строки, кортежи, множества и даже другие списки. Рассмотрим подробнее списки, элементами которых являются другие (вложенные) списки.

Практическое занятие: Решение задач по теме урока.

Тема 2.11 Знакомство со словарями (лекций 1 час, практических 1 час)

Лекция: Знакомство со словарями — встроенной в Python мощной структуре данных. В других языках аналогичная структура называется map, HashMap, Dictionary.

Базовые функции работы со словарями показаны на простых примерах хранения библиотеки знаний о фильмах и актерах.

Практическое занятие: Решение задач по теме урока.

Тема 2.12 Резерв (подготовка к КР/повторение) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала. Разбор с учениками задач из классных и домашних работ по темам из будущей КР.

Практическое занятие: Решение задач по пройденным темам.

Тема 2.13 Контрольная работа №1 (лекций 0 часов, практических 2 часа)

Практическое занятие: Решение задач из самостоятельной работы, которая состоит из четырех задач, которые можно решать в любом порядке.

Тема 2.14 Резерв (дорешка/tkinter/разбор КР) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала / Разбор с учениками задач из классных и домашних работ, разбор задач из КР / Рассматривается как писать небольшие программы, которые работают не с консолью, а с окном приложения (графическим пользовательским интерфейсом, GUI). Для этого нам понадобится дополнительная функциональность — библиотека tkinter. То, чем мы будем заниматься — только самое начало программирования GUI.

Практическое занятие: Решение задач по пройденным темам / Решение задач по теме урока.

Тема 2.15 Резерв (дорешка) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала, разбор с учениками задач из классных и домашних работ.

Практическое занятие: Решение задач по пройденным темам.

Задания для самостоятельной работы по модулю: Решение задач по пройденным темам.

Перечень ресурсов: Учебные материалы и задания по темам на платформе LMS Лицея Академии Яндекса <https://lyceum.yandex.ru/>.

3.3 Программа модуля 3 «Решение прикладных задач в Python»

Образовательная задача модуля:

Понятие ассоциативного массива. Словари в Python. Решение задач.

Модули в Python. Подключение и использование модулей стандартной библиотеки. Модульный принцип компоновки программы. Работа с документацией в стандартной библиотеке. Понятие репозитория различных пакетов Python. Работа с внешними библиотеками Python и утилитой pip. Основы ООП. Решение задач.

Самостоятельная работа: работа с конспектом, который описывает способы решения задач, разработка проекта по индивидуальному заданию, составление отчёта о выполнении индивидуальной или совместной работы.

Объем дисциплины (модуля): 76 часов,

в т.ч. лекций – 29 часов;

практических – 35 часов;

самостоятельной работы – 12 часов.

Содержание модуля

Тема 3.1 Функции (лекций 1 час, практических 1 час)

Лекция: Группирование команд в функции — участки кода, которые можно использовать многократно. Обсудим, как можно сделать так, чтобы код функции работал по-разному в зависимости от параметров. Наконец, коснемся вопроса, что представляют собой локальные переменные.

Практическое занятие: Решение задач по теме урока.

Тема 3.2 Функции. Возвращение значений из функции (лекций 1 час, практических 1 час)

Лекция: Обсудим, почему функции называются функциями, чем они похожи и чем отличаются от функций в математике. И главное: мы разберемся с основной целью вызова функций — возвратом во внешнюю программу результата вычисления.

Практическое занятие: Решение задач по теме урока.

Тема 3.3 Области видимости переменных (лекций 1 час, практических 1 час)

Лекция: Рассмотрим понятие «области видимости» переменных. Занятие будет посвящено в первую очередь переменным и лишь во вторую — функциям. Эти знания нечасто нужны для написания кода, но совершенно необходимы для понимания того, как программа работает. Это поможет вам не гадать, как ведет себя программа, и значительно сократит время, которое вы тратите на отладку.

Практическое занятие: Решение задач по теме урока.

Тема 3.4 Функции: передача параметров (лекций 1 час, практических 1 час)

Лекция: Рассмотрим, чем отличаются понятия «переменная» и «значение», ведь при вызове функций одно постоянно превращается в другое. Мы обсудим, почему аргумент, переданный в функцию, нельзя перезаписать, но иногда можно изменить его содержимое. Обсудим, какие объекты изменить можно, а какие не получится. Разберемся, почему равенство объектов неэквивалентно их идентичности.

Практическое занятие: Решение задач по теме урока.

Тема 3.5 Функции с переменным числом аргументов (лекций 1 час, практических 1 час)

Лекция: Научимся писать сложные функции, принимающие неопределенное число аргументов, и передавать в функцию именованные параметры. Попутно мы немного поговорим о том, как работает множественное присваивание.

Практическое занятие: Решение задач по теме урока.

Тема 3.6 Функции как объект. Лямбда-функции (лекций 1 час, практических 1 час)

Лекция: В языке Python все является объектом. Даже функция. Рассмотрим, как получить и использовать соответствующий объект. Кроме того, научимся создавать и использовать крошечные, буквально в полстроки функции и передавать их в качестве аргумента другой функции. Узнаем о двух важных функциях: `filter` и `map`, которые позволяют проводить множество преобразований над коллекциями и другими наборами объектов. Повторим списочные выражения и расширим знания о них.

Практическое занятие: Решение задач по теме урока.

Тема 3.7 Обработка коллекций. Поточковый ввод `sys.stdin` (лекций 1 час, практических 1 час)

Лекция: В Python встроено множество функций, которые помогают перебирать и комбинировать данные любыми способами. Ранее мы познакомились с функциями высшего порядка и даже попробовали их комбинировать. В Python нередко можно сложное вычисление свести к одной строке, если правильно подобрать порядок преобразования данных. В этом уроке будем изучать арсенал имеющихся инструментов и учиться их использовать.

Практическое занятие: Решение задач по теме урока.

Тема 3.8 Резерв (дорешка/итераторы и коллекции) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала, разбор с учениками задач из классных и домашних работ. / Дополнительные материалы посвящены более подробному изучению темы итераторов и их отличий от коллекций. Мы поговорим, как при помощи итераторов выполнять ленивые вычисления, позволяющие работать даже с бесконечными коллекциями объектов.

Хотя мы не собираемся рассматривать все возможности итераторов Python, мы начнем знакомство с модулем `itertools`. И разберем функцию свертки и группировки итераторов.

Практическое занятие: Решение задач по пройденным темам. / Решение задач по теме урока.

Тема 3.9 Резерв (дорешка/ переопределение функции и декораторы) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала, разбор с учениками задач из классных и домашних работ. / Из предыдущих занятий вспомним, что функцию можно записать в переменную. Оказывается, помимо этого, с именем функции можно работать как с самой обыкновенной переменной. И значение, на которое указывает переменная с именем функции, можно изменить. Изучим декораторы потому, что обычно они добавляют какие-то штрихи (декор) к уже существующему поведению функции, не изменяя ее код.

Практическое занятие: Решение задач по пройденным темам. / Решение задач по теме урока.

Тема 3.10 Самостоятельная работа №2 (лекций 0 часов, практических 2 часа)

Практическое занятие: Решение задач из самостоятельной работы, которая состоит из трёх задач, которые можно решать в любом порядке.

Тема 3.11 Резерв (Разбор СР/ДОП - Рекурсия) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала, разбор с учениками задач из самостоятельной работы. / Сегодня мы познакомимся с понятием рекурсии, покажем ее связь с уже известными нам конструкциями (циклами и функциями). Разберем наиболее часто встречающиеся ошибки и классические примеры.

Практическое занятие: Решение задач из СР. / Решение задач по теме урока.

Тема 3.12 Библиотеки Python. Часть №1 (Модули стандартной библиотеки) (лекций 1 час, практических 1 час)

Лекция: Python — высокоуровневый язык программирования, объектно-ориентированный, модульный и подчеркнута легко читаемый, что делает его очень простым в изучении. Python широко применяется в образовательной сфере, для научных вычислений, больших данных и машинного обучения, в веб-разработке, графике, GUI, играх и других направлениях. В связи с огромной сферой применения существует уже бесчисленное количество библиотек, упрощающих программирование на этом языке без необходимости написания излишнего кода. На этом уроке знакомимся со стандартными библиотеками языка Python.

Практическое занятие: Решение задач по теме урока.

Тема 3.13 Библиотеки Python. Часть №2 (графика) (Модули стандартной библиотеки) (лекций 1 час, практических 1 час)

Лекция: Ранее мы упоминали PyPI — кладезь библиотек для задач из разных областей. Обработка изображений — одна из таких областей, причем довольно обширная. С ней мы на этом занятии и познакомимся — у такого выбора есть целых три причины.

Практическое занятие: Решение задач по теме урока.

Тема 3.14 Библиотеки Python. Часть №3 (графика + звук) (лекций 1 час, практических 1 час)

Лекция: Продолжим изучение библиотеки PIL и научимся создавать собственные фильтры для растровых изображений. Кроме того, познакомимся с модулем wave для обработки звука.

Практическое занятие: Решение задач по теме урока.

Тема 3.15 Библиотеки Python. Часть №4 (морфология) (лекций 1 час, практических 1 час)

Лекция: Повторим что в Python имеется множество библиотек, реализующих абстракции для различных предметных областей. Поэтому программист занимается в основном верхним «этажом» логики. На этом занятии речь пойдет о библиотеке, позволяющей работать с текстами.

Практическое занятие: Решение задач по теме урока.

Тема 3.16 Библиотеки Python. Часть №5 (документы) (лекций 1 час, практических 1 час)

Лекция: Поговорим о библиотеках для работы с документами (текстами, презентациями, таблицами). Эти библиотеки можно использовать для автоматизации документооборота. Например, можно формировать документы: макеты презентаций, налоговую отчетность, открытки с поздравлениями.

Практическое занятие: Решение задач по теме урока.

Тема 3.17 Резерв (подготовка к СР/ДОП - NumPy) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала. Разбор с учениками задач из классных и домашних работ по темам из будущей СР. / Этот урок посвящен вычислительным мощностям Python и библиотеке NumPy. Научимся замерять время выполнения программы. Затронем работу с многомерными массивами и изучим (или вспомним) немного линейной алгебры. А еще, возможно, напишем игру.

Практическое занятие: Решение задач по пройденным темам. / Решение задач по теме урока.

Тема 3.18 Самостоятельная работа №3 (лекций 0 час, практических 2 часа)

Практическое занятие: Решение задач из самостоятельной работы, которая состоит из трёх задач, которые можно решать в любом порядке.

Тема 3.19 Введение в ООП (лекций 1 час, практических 1 час)

Лекция: Одна из самых распространенных методик разработки программных продуктов — объектно-ориентированное программирование (ООП). При работе с функциями мы уже использовали принцип модульности (сокрытие сложного алгоритма за вызовом функций) и библиотеки (упаковка функций, решающих схожие задачи, в одно хранилище — библиотеку). В ООП появляется еще один пример модульности — объект. Объекты хранят внутри себя и данные, и обрабатывающие их функции. Изучению парадигмы объектно-ориентированного программирования и посвящены следующие уроки.

Практическое занятие: Решение задач по теме урока.

Тема 3.20 Введение в ООП. Полиморфизм (лекций 1 час, практических 1 час)

Лекция: Рассмотрим возможности предоставления одинаковых средств взаимодействия с объектами разной природы.

Практическое занятие: Решение задач по теме урока.

Тема 3.21 Резерв (дорешка) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала, разбор с учениками задач из классных и домашних работ.

Практическое занятие: Решение задач по пройденным темам.

Тема 3.22 Резерв (дорешка) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала, разбор с учениками задач из классных и домашних работ.

Практическое занятие: Решение задач по пройденным темам.

Тема 3.23 ООП. Определение операторов (лекций 1 час, практических 1 час)

Лекция: Почти любой оператор Python можно определить и для типов данных, которые мы сами создаем с помощью классов. Это делается с помощью специальных методов. О них и пойдет речь.

Практическое занятие: Решение задач по теме урока.

Тема 3.24 ООП. Наследование (лекций 1 час, практических 1 час)

Лекция: Рассмотрим понятие наследования, покажем его связь с уже известными понятиями объектно-ориентированного программирования (инкапсуляцией и полиморфизмом), рассмотрим синтаксис и семантику наследования в Python.

Практическое занятие: Решение задач по теме урока.

Тема 3.25 ООП. Наследование. Часть №2 (лекций 1 час, практических 1 час)

Лекция: Этот урок посвящён приёмам применения наследования: расширению и переопределению методов. Обсуждается и множественное наследование.

Практическое занятие: Решение задач по теме урока.

Тема 3.26 Проектирование и разработка классов. Часть №1 (лекций 1 час, практических 1 час)

Лекция: Рассмотрим обширную тему проектирования программ. Чем больше и сложнее программа, тем важнее еще до написания кода понять, что она должна делать и какова будет ее внутренняя структура.

Практическое занятие: Решение задач по теме урока.

Тема 3.27 Проектирование и разработка классов. Часть №2 (лекций 1 час, практических 1 час)

Лекция: Продолжим проектирование и реализацию шахмат, которые не закончили в прошлый раз. Попробуем реализовать движение фигур по доске, на которой есть другие фигуры. Например, ладья, как и другие фигуры (за исключением коня), не может двигаться сквозь другие фигуры.

Практическое занятие: Решение задач по теме урока.

Тема 3.28 Резерв (подготовка к КР) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала. Разбор с учениками задач из классных и домашних работ по темам из будущей КР.

Практическое занятие: Решение задач по пройденным темам.

Тема 3.29 Контрольная работа №2 (лекций 0 часов, практических 2 часа)

Практическое занятие: Решение задач из самостоятельной работы, которая состоит из четырёх задач, которые можно решать в любом порядке.

Тема 3.30 Резерв (дорешка/разбор КР) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала / Разбор с учениками задач из классных и домашних работ, разбор задач из КР.

Практическое занятие: Решение задач по пройденным темам / Решение задач из КР.

Тема 3.31 Резерв (дорешка) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала, разбор с учениками задач из классных и домашних работ.

Практическое занятие: Решение задач по пройденным темам.

Тема 3.32 Резерв (дорешка) (лекций 1 час, практических 1 час)

Лекция: Повтор пройденного материала, разбор с учениками задач из классных и домашних работ.

Практическое занятие: Решение задач по пройденным темам.

Задания для самостоятельной работы по модулю: Решение задач по пройденным темам.

Перечень ресурсов Учебные материалы и задания по темам на платформе LMS Лицея Академии Яндекса <https://lyceum.yandex.ru/>.

4. Календарный учебный график дополнительной общеобразовательной общеразвивающей программы «Основы программирования на языке Python (углубленный уровень)»

Начало и окончание реализации программы: октябрь – апрель.

Каникулы: не предусмотрены.

Продолжительность занятий: 45 минут

Продолжительность перемен: 10 минут

Образовательная недельная нагрузка на обучающихся: ~ 5 часов.

Наименование модуля (раздела) / темы	Неделя	Кол-во часов
Модуль 1. «Введение в программирование»		
Тема 1.1 Знакомство со средой	1	3
Тема 1.2 Условный оператор	1	3
Тема 1.3 Простые встроенные функции	2	3
Тема 1.4 Знакомство с циклом while	2	3
Тема 1.5 Отладчик	3	3
Тема 1.6 Знакомство с циклом for	3	3
Тема 1.7 true и False, break и continue	4	3
Тема 1.8 Вложенные циклы	4	3
Тема 1.9 Резерв (дорешка)	5	2
Тема 1.10 Резерв (дорешка)	5	2
Тема 1.11 Резерв (пробная СР/разбор пробной СР)	6	2
Модуль 2. «Базовые конструкции в Python»		
Тема 2.1 Множества	6	3
Тема 2.2 Строки. Индексация	7	3
Тема 2.3 Строки. Срезы	7	3

Наименование модуля (раздела) / темы	Неделя	Кол-во часов
Тема 2.4 Знакомство со списками	8	2
Тема 2.5 Кортежи. Преобразование коллекций	8	3
Тема 2.6 Методы split и join. Списочные выражения	9	3
Тема 2.7 Методы списков и строк	9	3
Тема 2.8 Резерв (подготовка к СР/повторение)	10	2
Тема 2.9 Самостоятельная работа №1	10	2
Тема 2.10 Вложенные списки	11	3
Тема 2.11 Знакомство со словарями	11	3
Тема 2.12 Резерв (подготовка к КР/повторение)	12	2
Тема 2.13 Контрольная работа №1	12	2
Тема 2.14 Резерв (дорешка/tkinter/разбор КР)	13	2
Тема 2.15 Резерв (дорешка)	13	2
Модуль 3. «Решение прикладных задач в Python»		
Тема 3.1 Функции	14	2
Тема 3.2 Функции. Возвращение значений из функции	14	2
Тема 3.3 Области видимости переменных	15	3
Тема 3.4 Функции: передача параметров	15	3
Тема 3.5 Функции с переменным числом аргументов	16	3
Тема 3.6 Функции как объект. Лямбда-функции	16	3
Тема 3.7 Обработка коллекций. Поточковый ввод sys.stdin	17	2
Тема 3.8 Резерв (дорешка/итераторы и коллекции)	17	2
Тема 3.9 Резерв (дорешка/ переопределение функции и декораторы)	18	2
Тема 3.10 Самостоятельная работа №2	18	2
Тема 3.11 Резерв (Разбор СР/ДОП - Рекурсия)	19	2
Тема 3.12 Библиотеки Python. Часть №1 (Модули стандартной библиотеки)	19	3
Тема 3.13 Библиотеки Python. Часть №2 (графика)	20	3

Наименование модуля (раздела) / темы	Неделя	Кол-во часов
Тема 3.14 Библиотеки Python. Часть №3 (графика + звук)	20	3
Тема 3.15 Библиотеки Python. Часть №4 (морфология)	21	3
Тема 3.16 Библиотеки Python. Часть №5 (документы)	21	3
Тема 3.17 Резерв (подготовка к СР/ДОП - NumPy)	22	2
Тема 3.18 Самостоятельная работа №3	22	2
Тема 3.19 Введение в ООП	23	2
Тема 3.20 Введение в ООП. Полиморфизм	23	2
Тема 3.21 Резерв (дорешка)	24	2
Тема 3.22 Резерв (дорешка)	24	2
Тема 3.23 ООП. Определение операторов.	25	2
Тема 3.24 ООП. Наследование	25	2
Тема 3.25 ООП. Наследование. Часть №2	26	3
Тема 3.26 Проектирование и разработка классов. Часть №1	26	3
Тема 3.27 Проектирование и разработка классов. Часть №2	27	3
Тема 3.28 Резерв (подготовка к КР)	27	2
Тема 3.29 Контрольная работа №2	28	2
Тема 3.30 Резерв (дорешка/разбор КР)	28	2
Тема 3.31 Резерв (дорешка)	29	2
Тема 3.32 Резерв (дорешка)	29	2

5. Организационно-педагогические условия реализации программы

5.1 Кадровое обеспечение

Название дисциплины / модуля / практики	ФИО преподавателя	Квалификация преподавателей (образование, ученая степень, ученое звание, награды, звания); квалификация преподавателей, привлекаемых к проведению занятий	Опыт профессиональной деятельности (преподавательской деятельности) (стаж работы)
«Основы программирования на языке Python (углубленный уровень)»	Попов Александр Александрович	Образование – высшее, инженер по специальности промышленная электроника. Педагог дополнительного образования детей ДНК им. Академика Е. М. Дианова, инженер научно-исследовательской лаборатории вентильных электрических машин, младший научный сотрудник Института корпоративного обучения и непрерывного образования	5
«Основы программирования на языке Python	Яшин Евгений Евгеньевич	Образование – высшее.	5

(углубленный уровень)»			
------------------------	--	--	--

5.2 Учебно-методическое и информационное обеспечение

Основная

1. К. Ю. Поляков, Е. А. Еремин. Информатика. Углублённый уровень. Учебник для 10 класса в 2 частях. М.: БИНОМ. Лаборатория знаний, 2014.
2. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.
3. Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ. Лаборатория знаний, 2006.
4. С. М. Окулов. Основы программирования. М.: Бином. Лаборатория знаний, 2012.

Литература, рекомендованная обучающимся

5. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.
6. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И. Г. Семакина и Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.

Электронные ресурсы:

1. Материалы и презентации к урокам в LMS Яндекс.Лицея.
2. Сайт pythonworld.ru — «Python 3 для начинающих».
3. Сайт pythontutor.ru — «Питонтьютор».

<https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwBl> — Лекции А.В. Умнова, прочитанные в Школе Анализа Данных Яндекса.

5.3 Материально-техническое обеспечение

Наименование специализированных учебных помещений	Вид занятий	Наименование оборудования, программного обеспечения
1	2	3
Аудитория 401, 201	Лекция, практика	Ноутбуки с доступом к сети интернет, операционная система Windows, программное обеспечение PyCharm, наличие доступа к сети Интернет, наличие кабинета

6. Нормативно-методическое обеспечение системы оценки качества освоения обучающимися дополнительной общеобразовательной общеразвивающей программы

6.1 Формы аттестации

Название	Форма аттестации
Модуль 1 «Введение в программирование»	Пробная самостоятельная работа
Модуль 2 «Базовые конструкции в Python»	Самостоятельная работа, контрольная работа
Модуль 3 «Решение прикладных задач в Python»	Самостоятельная работа, контрольная работа
Итоговая аттестация	Сумма баллов за курс

6.2 Оценочные материалы

6.2.1 Комплект оценочных средств для проведения промежуточной аттестации

Модуль	Перечень контрольных заданий (вопросов, практических заданий, задач, ситуаций, тестовых заданий и др.)	Критерии оценки
Модуль 1 «Введение в программирование»	3-4 задачи на пройденные темы модуля, с 2-3 тестами для самостоятельной проверки правильности решения.	Проверка программного кода на соответствие стандарту PEP8, количеству пройденных автотестов в тестирующей системе.
Модуль 2 «Базовые конструкции в Python»	3-4 задачи на пройденные темы модуля, с 2-3 тестами для самостоятельной проверки правильности решения.	Проверка программного кода на соответствие стандарту PEP8,

		количеству пройденных автотестов в тестирующей системе.
Модуль 3 «Решение прикладных задач в Python»	3-4 задачи на пройденные темы модуля, с 2-3 тестами для самостоятельной проверки правильности решения.	Проверка программного кода на соответствие стандарту PEP8, количеству пройденных автотестов в тестирующей системе.

6.2.2 Комплект оценочных средств для проведения итоговой аттестации

Рейтинг — это показатель успешности ученика в Лицее Академии Яндекса. Рейтинг обновляется с каждой зачтенной задачей. Стоимость задачи назначает методист курса.

Максимальный рейтинг — 120 баллов. Чтобы перейти на второй курс, нужно набрать 45 и более баллов.

Соответствие рейтинга и оценки:

Рейтинг	Оценка
<45	Удовлетворительно
45–79	Хорошо
80–99	Отличник
100 и больше	Суперотличник

Как вычисляется рейтинг:

Шаг 1. Устанавливаются максимально возможные баллы за каждый тип задач:

Все задачи курса делятся на пять типов: домашние, дополнительные и задачи классных, самостоятельных, контрольных работ.

Максимальные баллы:

- 10 баллов за решение всех задач классных работ ($MAX_{кл}$),
- 10 баллов — домашних задач ($MAX_{дом}$)
- 40 баллов — дополнительных задач ($MAX_{дон}$)
- 40 баллов — задач контрольных работ ($MAX_{контр}$)
- 20 баллов — задач самостоятельных работ ($MAX_{сам}$).



Решение дополнительных задач, задач самостоятельных и контрольных работ вносит самый большой вклад в рейтинг:

Шаг 2. Вычисляется количество уроков с задачами каждого типа

В курсах может быть разное количество задач каждого типа. Например, на первом курсе основного образования ученики решают четыре самостоятельные работы, а на первом курсе дополнительного образования — три.

Поэтому для каждого типа задач рассчитывается K_{min} — количество уроков курса, в которых есть хотя бы одна задача данного типа.

Например, $K_{min} = 2$ означает, что в курсе есть два урока хотя бы с одной задачей классной работы.

Шаг 3. Суммируются первичные баллы решенных задач

Первичные баллы — это баллы, указанные в тестирующей системе для каждой задачи.

Попугай

Классная работа макс. 16 балл. Дедлайн: 1 окт. 23:30

Условие

ID 32524072 Доработать

25 апр. 12:57 25 апр. 12:58

```
1 print('Моя первая программа напечатала "Привет, Яндекс!" :')
2 sdfsd
3 file()sdf
4
5 sdf
6 sd
7 file()sd
8 file()sdf
9
10 sdf
```

Отправить решение

История решений

ID 21031080
Отправлено на проверку

Дедлайн: 1 окт. 23:30

ID 32524072
Отправлено на проверку

Вердикт: ошибка оформления кода. Подробнее

Статус изменен: Доработать

Введем обозначения:

- $ПБ_{кл} = \sum ПБ_{РешенныхКлассных}$
- $ПБ_{дом} = \sum ПБ_{РешенныхДомашних}$
- $ПБ_{доп} = \sum ПБ_{РешенныхДополнительных}$
- $ПБ_{контр} = \sum ПБ_{РешенныхКонтрольных}$
- $ПБ_{сам} = \sum ПБ_{РешенныхСамостоятельных}$

За выполнение всех задач одного типа дается максимум 100 первичных баллов. Например, если ученик решит все домашние задачи урока, то $ПБ_{дом} = 100$.

Шаг 4. Вычисляются вторичные баллы

Вторичные баллы вычисляются по формуле:

$$ВБ_{ТипЗадачи} = \frac{ПБ_{ТипЗадачи}}{100} \cdot \frac{МАХ_{ТипЗадачи}}{K_{ТипЗадачи}}$$

Первый множитель в формуле — доля набранных первичных баллов из максимально возможных. Второй множитель — стоимость одного урока в баллах.

Введем обозначения:

$$\bullet \text{ ВБ}_{\text{кл}} = \frac{\text{ПБ}_{\text{кл}}}{100} \cdot \frac{\text{МАХ}_{\text{кл}}}{K_{\text{кл}}}$$

$$\bullet \text{ ВБ}_{\text{дом}} = \frac{\text{ПБ}_{\text{дом}}}{100} \cdot \frac{\text{МАХ}_{\text{дом}}}{K_{\text{дом}}}$$

$$\bullet \text{ ВБ}_{\text{доп}} = \frac{\text{ПБ}_{\text{доп}}}{100} \cdot \frac{\text{МАХ}_{\text{доп}}}{K_{\text{доп}}}$$

$$\bullet \text{ ВБ}_{\text{контр}} = \frac{\text{ПБ}_{\text{контр}}}{100} \cdot \frac{\text{МАХ}_{\text{контр}}}{K_{\text{контр}}}$$

$$\bullet \text{ ВБ}_{\text{сам}} = \frac{\text{ПБ}_{\text{сам}}}{100} \cdot \frac{\text{МАХ}_{\text{сам}}}{K_{\text{сам}}}$$

Шаг 5. Вторичные баллы суммируются

Рейтинг вычисляется по формуле:

$$P = \text{ВБ}_{\text{кл}} + \text{ВБ}_{\text{дом}} + \text{ВБ}_{\text{доп}} + \text{ВБ}_{\text{контр}} + \text{ВБ}_{\text{сам}}$$

7. Сведения об обновлении программы

Программа обновлена решением Ученого совета Университета:

№	Прилагаемый к ДООП документ, содержащий текст обновления	Решение об обновлении ДООП	
		дата	протокол №
1.	Приложение № 1	____. ____ 20__ г.	
2.	Приложение № 2	____. ____ 20__ г.	
3.	Приложение № 3	____. ____ 20__ г.	
4.	Приложение № 4	____. ____ 20__ г.	