



«Рассмотрено»  
Руководитель ТМО  
 /Валиева В.З.

«Согласовано»  
Заместитель директора по УР  
МБОУ лицей №2  
 /Лунева И.А.

«Утверждаю»  
Директор МБОУ  
лицей №2  
 /Иванов Г.А.  
Приказ № 100 от  
«31» августа 2023 г.

Муниципальное бюджетное общеобразовательное учреждение лицей №2  
Бугульминского муниципального района Республики Татарстан

**РАБОЧАЯ ПРОГРАММА**  
кружка Program Планета

Направленность Техническая  
Уровень образования 8-9 классы  
Срок реализации 2023-2024 учебный год

Составитель Акимова Т.Е.  
Учитель информатики  
Первой квалификационной категории

Бугульма, 2023/2024 учебный год

## Оглавление

Пояснительная записка.....	3
Нормативно-правовое обеспечение.....	3
Актуальность программы.....	4
Направленность программы.....	4
Общепедагогическая направленность занятий .....	4
Цели изучения курса:.....	4
Задачи курса.....	4
Адресат общеразвивающей программы.....	5
Возрастные особенности группы.....	5
Методы обучения.....	5
Формы организации учебных занятий .....	5
Планируемые результаты курса.....	5
Личностные: .....	6
Метапредметные:.....	6
Способы отслеживания результатов освоения программы учащимися: .....	7
Способы оценивания уровня достижений учащихся .....	7
Учебно-тематический план .....	8
Содержание тем учебного курса .....	11
Тема 1. Знакомство с языком Python.....	11
Тема 2. Переменные и выражения.....	11
Тема 3. Условные предложения.....	12
Тема 4. Циклы.....	12
Тема 5. Функции.....	13
Тема 6. Строки - последовательности символов.....	13
Тема 7. Сложные типы данных.....	14
Тема 8. Стиль программирования и отладка программ .....	15
Литература и источники.....	16

## Пояснительная записка

Данный курс «ProgramПланета» для внеурочной деятельности и рассчитан на 68 часов изучения языка Python. Содержание обучения, представленное в программе курса изучения Python, позволяет вести обучение школьников в режиме актуального познания.

Направленность курса – техническая. Практическая направленность курса на создание внешних образовательных продуктов — блок-схем, алгоритмов, программ — способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у школьников знаний. Возникающие при этом познавательные переживания обуславливают сознательное отношение к изучению основных теоретических положений информатики.

Проявления трудолюбия, целеустремленности, возникающие при воплощении замыслов учащихся, в рамках курс «ProgramПланета», стимулируют развитие индивидуально-личностных качеств школьников.

Активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению освоенных навыков программирования в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся.

## Нормативно-правовое обеспечение

Программа разработана в соответствии:

- ~ Методическими рекомендациями по проектированию и реализации дополнительных общеобразовательных программ (в том числе адаптированных) в новой редакции. Письмо МОиН РТ №2749/23 от 07.03.2023
- ~ С Федеральным законом от 29.12.2012 №273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями, вступившими в силу с 29.12.2022
- ~ Федеральным законом от 31.07.2020 №304 ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации» по вопросам воспитания обучающихся»
- ~ Приказом Министерства просвещения Российской Федерации от 27.07.2022г №629 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»
- ~ С учетом положений Концепции развития дополнительного образования детей, утвержденной Распоряжением Правительства РФ от 31.03.2022 №678-р.
- ~ Целевой модели развития региональных систем дополнительного образования детей, утвержденной Приказом Министерства просвещения Российской Федерации от 3.09.2019 №467
- ~ Федеральный закон от 13 июля 2020г. № 189-ФЗ «О государственном (муниципальном) социальном заказе на оказание государственных (муниципальных) услуг в социальной сфере» (с изменениями и дополнениями, вступившими в силу с 28.12.2022г.)
- ~ «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», утвержденные Постановлением Главного государственного санитарного врача Российской Федерации от 28.09.2020г. №28
- ~ Уставом муниципального бюджетного общеобразовательного учреждения МБОУ лицей №2

- Положением о рабочей программе муниципального бюджетного общеобразовательного учреждения МБОУ лицей №2, и иных нормативных правовых документов.

### Актуальность программы.

С 1996 г., популярность языка Python неуклонно растёт. Это отражено в рейтинге популярных языков программирования по индексу ТЮВЕ — в нём питон занимает третье место после языков С и Java. Python — высокоуровневый язык общего назначения. Это значит, что такой язык быстр и удобен в работе, а программы, написанные на нём, просты для понимания программистами. Обычно его используют для веб-разработки, анализа данных, написания скриптов и игр, а также искусственного интеллекта.

### Направленность программы

Курс «ProgramПланета» имеет техническую направленность, что позволяет обучающимся приобщиться к инженерно-техническим знаниям в области инновационных технологий, сформировать техническое мышление. Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и реализации алгоритма с помощью средств программирования.

**Общепедагогическая направленность занятий** – гармонизация индивидуальных и социальных аспектов обучения по отношению к информационным технологиям. Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции — одной из ключевых компетенций современной школы. Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных школьников. Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

В рамках предлагаемого курса изучение основ программирования на языке Python — это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Исключительно велика роль программирования для формирования мышления школьников, приёмов умственных действий, умения строить модели, самостоятельного нахождения и составления алгоритмов решения задач, умения чётко и лаконично реализовывать этапы решения задач. Использование этих возможностей для формирования общеинтеллектуальных и общеучебных умений школьников активизирует процесс индивидуально-личностного становления учащихся.

### Цели изучения курса:

- понять значение алгоритмизации как метода познания окружающего мира, принципы структурной алгоритмизации;
- овладеть базовыми понятиями теории алгоритмов;
- научиться разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

### Задачи курса:

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- научить составлять и читать блок-схемы;

- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

### Адресат общеразвивающей программы

Программа изучения программирования на Python предназначена для детей в возрасте 12–17 лет, мотивированных к обучению и проявляющих интерес к IT-технологиям, приобретению навыков программирования. Формы занятий групповые. Количество обучающихся в группе – 12-15 человек. Состав групп постоянный.

### Возрастные особенности группы

Содержание программы учитывает возрастные и психологические особенности детей в возрасте 12–17 лет, которые определяют выбор форм проведения занятий с обучающимися. Подростки данной возрастной группы характеризуются такими процессами, как изменение структуры личности и бурного физического развития. Происходят качественные изменения и в познавательной деятельности, и в личности, и в межличностных отношениях. У каждого эти изменения происходят в разное время. В этом возрасте начинается переход от мышления, основанного на оперировании конкретными представлениями, к теоретическому мышлению, от непосредственной памяти к логической. В 12-14 лет ведущий тип деятельности – референтно значимый, к нему относятся: проектная деятельность (встреча замысла и результата как авторское действие подростка), проявление себя в общественно значимых ролях (выход в настоящую взрослую действительность).

12–14 лет – подростковый период. Характерная особенность – личное самосознание, сознательное проявление индивидуальности. Ведущая потребность – самоутверждение. В подростковый период стабилизируются интересы детей. Основное новообразование – становление взрослости как стремление к жизни в обществе взрослых. Основная задача педагога в работе с детьми в возрасте 15–17 лет сводится к решению противоречия между готовностью их к полноценной социальной жизни и недопущением отставания от жизни содержания и организации их образовательной деятельности. Также следует отметить, что подростки в возрасте 15-17 лет характеризуются такими психическими процессами, как стремление углублённо понять себя, разобраться в своих чувствах, настроениях, мнениях, отношениях.

### Методы обучения

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности учащихся, реализовывать личностно-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности. Решение данных задач кроется в организации деятельностного подхода к обучению, в проблемном изложении материала учителем, в переходе от репродуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном курсе — обучение учащихся навыкам самостоятельной творческой деятельности.

### Формы организации учебных занятий

Организация учебного процесса предусматривает дистанционную формы деятельности, когда учащийся вне уроков самостоятельно выполняет на компьютере практические задания.

### Планируемые результаты курса

В рамках курс «Языки программирования. Python» учащиеся овладевают следующими знаниями, умениями и способами деятельности:

- умеют составлять алгоритмы для решения задач;
- умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владеют основными навыками программирования на языке Python;
- умеют отлаживать и тестировать программы, написанные на языке Python.

#### Личностные:

- широкие познавательные интересы, инициатива и любознательность, мотивы познания и творчества; готовность и способность учащихся к саморазвитию и реализации творческого потенциала в духовной и предметно-продуктивной деятельности за счет развития их образного, алгоритмического и логического мышления;
- готовность к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов информатики и ИКТ;
- интерес к информатике, алгебре, геометрии и физике, стремление использовать полученные знания в процессе обучения другим предметам и в жизни;
- способность связать учебное содержание с собственным жизненным опытом и личными смыслами, понять значимость подготовки в области информатики и ИКТ в условиях развития информационного общества;
- готовность к самостоятельным поступкам и действиям, принятию ответственности за их результаты;
- владение основными универсальными умениями информационного характера: постановка и формулирование проблемы;
- поиск и выделение необходимой информации, применение методов информационного поиска; умение использовать линейные, разветвляющиеся и циклические операторы при создании программы на языке программирования Python;
- умение использовать логические значения, операции и выражения с ними;
- умение формально выполнять алгоритмы, описанные с использованием конструкций ветвления и повторения, вспомогательных алгоритмов;
- способность к избирательному отношению к получаемой информации за счет умений ее анализа и критичного мышления; – способность и готовность к принятию ценностей здорового образа жизни за счет знания основных условий безопасной эксплуатации средств ИКТ;
- возможность определиться при выборе будущей профессии.

#### Метапредметные:

- владение умениями организации собственной учебной деятельности, а именно: постановку учебной задачи на основе того, что уже известно, и того, что требуется установить;
- планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;
- контроль – интерпретация полученного результата, его соотнесение с имеющимися данными с целью установления соответствия или несоответствия (обнаружения ошибки);
- коррекция – внесение необходимых дополнений и корректив в план действий в случае обнаружения ошибки;
- оценка – осознание учащимся того, насколько качественно им решена учебно-познавательная задача;
- умение выступать перед аудиторией, представляя ей результаты своей работы с помощью средств ИКТ;

- использование коммуникационных технологий в учебной деятельности и повседневной жизни.

#### Способы отслеживания результатов освоения программы учащимися:

- педагогическое наблюдение в ходе занятий;
- командные зачеты;
- участие в практикумах, мини-проектах;
- презентация своего мини-проекта на публику.

#### Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты учащихся (созданные блок-схемы, программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество внешней образовательной продукции желательно оценивать по следующим параметрам:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- программа должна выполнять поставленные задачи;
- по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).

Проверка достигаемых учащимися результатов производится в следующих формах:

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

Итоговый контроль проводится в конце всего курса. Он организуется в виде тестирования.



## Учебно-тематический план

Раздел или тема программы	Количество часов			Дата	Форма организации занятий	Форма контроля
	Всего	Теория	Практика			
<b>Тема 1. Знакомство с языком Python</b>	<b>4</b>	<b>1</b>	<b>3</b>		Выполнение лабораторных работ, практических задач, кейс-заданий	
<i>Урок 1. Общие сведения о языке</i>	2	1		4.09	беседа	
Практическая работа 1.1. Установка программы Python			1		Пр.р.	
<i>Урок 2. Режимы работы</i>	2			11.09	беседа	
Практическая работа 1.2. Режимы работы с Python			2		Пр.р.	
Тест № 1. Знакомство с языком Python					Инд работа	тест
<b>Тема 2. Переменные и выражения</b>	<b>8</b>	<b>3</b>	<b>5</b>		Выполнение лабораторных работ, практических задач, кейс-заданий	
<i>Урок 3. Переменные</i>	2			18.09	опрос	
Практическая работа 2.1. Работа со справочной системой			1		Пр.Р.	
Практическая работа 2.2. Переменные			1		Пр.р.	
<i>Урок 4. Выражения</i>	2	1		25.09	Груп. работа	
Практическая работа 2.3. Выражения			1		Пр.р.	
<i>Урок 5. Ввод и вывод</i>	2	1		2.10	Опрос	
Практическая работа 2.4. Ввод и вывод			1		Пр.р.	
<i>Урок 6. Задачи на элементарные действия с числами</i>	2	1		9.10	Груп.работа	
Практическая работа 2.5. Задачи на элементарные действия с числами			1		Пр.р.	
Тест № 2. Выражения и операции.					Инд.работа	Тест
<b>Тема 3. Условные предложения</b>	<b>10</b>	<b>2</b>	<b>8</b>		Выполнение лабораторных работ, практических задач, кейс-задани	
<i>Урок 7. Логические выражения и операторы</i>	2	1		16.10	Опрос	
Практическая работа 3.1. Логические выражения			1		Пр.р.	
<i>Урок 8. Условный оператор</i>	2			23.10	беседа	
Практическая работа 3.2. "Условный оператор"			2		Пр.р.	
<i>Урок 9. Множественное ветвление</i>	2	1		6.11	опрос	
Практическая работа 3.3. Множественное ветвление			1		Пр.р.	



Урок 10. Реализация ветвления в языке Python	2			13.11	беседа	
Практическая работа 3.4. "Условные операторы"			1		Пр.р.	
Самостоятельная работа № 1 по теме "Условные операторы".			1		Сам.работа	
Урок 11. Зачетная работа № 1. "Составление программ с ветвлением".	2		2	20.11	Инд.работа	Зачет
Тест № 3. "Условные операторы".					Инд.работа	Тест
<b>Тема 4. Циклы</b>	<b>14</b>	<b>2</b>	<b>12</b>		Выполнение лабораторных работ, практических задач, кейс-задани	
Урок 12. Оператор цикла с условием	2	1		27.11	Мозговой штурм	
Практическая работа 4.1. "Числа Фибоначчи"			1		Пр.р.	
Урок 13. Оператор цикла for	2			4.12	Беседа	
Практическая работа 4.2. Решение задачи с циклом for.			2		Пр.р.	
Урок 14. Вложенные циклы	2			11.12	Видеолекция	
Практическая работа 4.3. Реализация циклических алгоритмов			2		Пр.р.	
Урок 15. Случайные числа	2			18.12	Опрос	
Практическая работа 4.4. Случайные числа			2		Пр.р.	
Урок 16. Примеры решения задач с циклом	2			25.12	Практикум	
Практическая работа 4.5. Решение задач с циклом.			1		Пр.р.	
Тест № 4. Циклы		1			Инд.работа	тест
Самостоятельная работа № 2 "Составление программ с циклом"	2		2	15.01	Сам.работа	
Урок 17. Творческая работа № 1. "Циклы"	2		2	22.01	Практикум	
<b>Тема 5. Функции</b>	<b>10</b>	<b>2</b>	<b>8</b>		Выполнение лабораторных работ, практических задач, кейс-задани	
Урок 18. Создание функций	2			29.01	Беседа	
Практическая работа 5.1. Создание функций			2		Пр.р.	
Урок 19. Локальные переменные	2	1		5.02	Диспут	
Практическая работа 5.2. Локальные переменные			1		Пр.р.	
Урок 20. Примеры решения задач с использованием функций	2			12.02	Практикум	
Практическая работа 5.3.			2		Пр.р.	

Решение задач с использованием функций						
Самостоятельная работа № 3 по теме "Функции"	2		2	19.02	Сам.работа	
Урок 21. Рекурсивные функции	2			26.02	Опрос	
Практическая работа 5.4. Рекурсивные функции			2		Пр.р.	
Тест № 5. Функции		1			Инд.работа	тест
<b>Тема 6. Строки - последовательности символов</b>	<b>6</b>	<b>1</b>	<b>5</b>		Выполнение лабораторных работ, практических задач, кейс-заданий	
Урок 22. Строки	2	1	1	4.03	Практикум	
Урок 23. Срезы строк	2			11.03	Практикум	
Практическая работа 6.1. Строки			2		Пр.р.	
Урок 24. Примеры решения задач со строками	2			18.03	Груп.работа	
Практическая работа 6.2. Решение задач со строками.			2		Пр.р.	
<b>Тема 7. Сложные типы данных</b>	<b>12</b>		<b>4</b>		Выполнение лабораторных работ, практических задач, кейс-заданий	
Урок 25. Списки	2	2		25.03	Практикум	
Урок 26. Срезы списков	2			1.04	Практикум	
Практическая работа 7.1. Списки			2		Пр.р.	
Урок 27. Списки: примеры решения задач	2			8.04	Практикум	
Практическая работа 7.2. Решение задач со списками			2		Пр.р.	
Урок 28. Матрицы	2	1		15.04	Практикум	
Тест № 7. Списки		1			Беседа	
Урок 29. Кортежи	2	1	1	22.04	Практикум	
Урок 30. Введение в словари	2	1	1	29.04	Практикум	
Урок 31. Множества в языке Python	2	1	1	6.05	Практикум	
<b>Тема 8. Стиль программирования и отладка программ</b>	<b>6</b>	<b>1</b>	<b>2</b>		Выполнение лабораторных работ, практических задач, кейс-заданий	
Урок 32. Стиль программирования	2	1	1	13.05	Практикум	
Урок 33. Отладка программ	2		2	20.05	Практикум	
Урок 34. Зачет по курсу «Программирование на языке Python»	2		2	27.05	Инд.работа	Защита проектной работы
Урок 35. Что дальше?		1			Мозговой штурм	
<b>ВСЕГО</b>	<b>68</b>	<b>21</b>	<b>49</b>			

# Содержание тем учебного курса

---

## Тема 1. Знакомство с языком Python

Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python.

Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python

Практическая работа 1.2. Режимы работы с Python

Тест № 1. Знакомство с языком Python Учащиеся

должны знать / понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.
- 

## Тема 2. Переменные и выражения

Типы данных . Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой

Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.4. Задачи на элементарные действия с числами

Тест № 2. Выражения и операции.

Учащиеся должны знать / понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- операторы ввода-вывода.

Учащиеся должны уметь:

- пользоваться интерфейсом среды программирования Python;
- использовать команды редактора;
- организовывать ввод и вывод данных;
- записывать арифметические выражения.

### Тема 3. Условные предложения

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения

Практическая работа 3.2. "Условный оператор"

Практическая работа 3.3. Множественное ветвление

Практическая работа 3.4. "Условные операторы"

Самостоятельная работа № 1. Решение задач по теме "Условные операторы".

Зачетная работа № 1. "Составление программ с ветвлением".

Тест № 3. "Условные операторы".

Учащиеся должны знать / понимать:

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;

Учащиеся должны уметь:

- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

### Тема 4. Циклы

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи"

Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов

Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом.

Самостоятельная работа № 2 "Составление программ с циклом"

Тест № 4. Циклы

Творческая работа № 1. "Циклы"

Учащиеся должны знать / понимать:

- циклы с условием и их виды;
- правила записи циклов с условием;
- назначение и особенности использования цикла с параметром;

- формат записи цикла с параметром;
- примеры использования циклов различных типов.

Учащиеся должны уметь:

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи;

## Тема 5. Функции

Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция `lambda`. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций

Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа № 3 по теме "Функции"

Тест № 5. Функции

Учащиеся должны знать / понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Учащиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

## Тема 6. Строки - последовательности символов

Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор `in`. Модуль `string`. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

Учащиеся должны знать / понимать:

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;

- операции со строками.

Учащиеся должны уметь:

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

## Тема 7. Сложные типы данных

Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков.

Срезы списков. Удаление списка. Клонирование списков. Списочные параметры.

Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками.

Тест № 7. Списки

Учащиеся должны знать / понимать:

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

Учащиеся должны уметь:

- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;

- выводить элементы множества.

## Тема 8. Стил ь программирования и отладка программ

Стил ь программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

Учащиеся должны знать / понимать:

- что такое стил ь программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.



## Литература и источники

1. Домашняя страница Python [www.python.org](http://www.python.org) . Справочные материалы, официальная документация.
2. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python»,  
  
<http://www.intuit.ru/studies/courses/12179/1172/info>.
3. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования *Python*» <http://www.intuit.ru/studies/courses/49/49/info>.
4. Сайт проекта Open Book Project [openbookproject.net](http://openbookproject.net) содержит серию практических примеров на Python Криса Мейерса.
5. *Python. Подробный справочник* Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.
6. *Python. Справочник* Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.