

**Муниципальное общеобразовательное бюджетное учреждение
средняя общеобразовательная школа с.Ямады муниципального района
Янаульский район Республики Башкортостан**

Рассмотрена на заседании
педагогического совета
протокол № 2
от «29» августа 2019 г.

«Утверждаю»:
Директор МБОУ СОШ с.Ямады
С.Р. Сафина
Приказ № 44 от «29» августа 2019 г.



**Дополнительная общеразвивающая программа
технического направления
«Язык программирования -Python»**

Возраст обучающихся: 13– 16 лет

Срок реализации: 1 год

**Составитель программы:
педагог дополнительного образования
Гумерова Р.А.**

I. Пояснительная записка

Дополнительная общеобразовательная программа «**Язык программирования - Python**» (далее – Программа) физкультурно-спортивной направленности муниципального общеобразовательного бюджетного учреждения средней общеобразовательной школы с.Ямады предназначена для занятий дополнительного образования в Центре образования цифрового и гуманитарного профилей «Точка роста» с детьми школьного возраста. Данный рассчитан на 64 часа.

Изменение взглядов на предмет информатики как науки, её место в системе научного знания требует существенных изменений в содержании образования по информатике. В связи с этим особую актуальность приобретают раскрытие личностных резервов учащихся и создание соответствующей среды.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если учащиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей, связанных с познанием, если они видят смысл в сотрудничестве с одноклассниками и учителем.

Содержание обучения, представленное в программе курс «Языки программирования. Python», позволяет вести обучение школьников в режиме актуального познания. 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 «Языки программирования. Python», 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.

Концепция курса

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.

рамках предлагаемого курса «Язык программирования - Python» 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.

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.

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 курса «Язык 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.

Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Язык программирования - 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 тестирования.

Программное обеспечение:

1. Операционная система: Windows XP (или выше).
2. Среда разработки: Python 3.3 (или выше),

II. ЦЕЛИ И ЗАДАЧИ

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.

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 форму деятельности, когда учащийся вне уроков самостоятельно выполняет на компьютере практические задания.

Планируемые результаты курса

В рамках курса «Язык 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.

Способы оценивания уровня достижений учащихся

Предметом диагностики и контроля в курсе «Язык программирования – 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.

III. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

Занятия проводятся один раз в неделю по 2,25 часа

	Наименование разделов и тем	Количество часов	Теория	Практика
1.	Знакомство с Python	4,5		
1.1	Техника безопасности. Общие сведения о языке.	2,25	1	1,25
1.2	Структура программы на языке Python	2,25	1	1,25
2	Основные алгоритмические конструкции	22,5		
2.1	Условный оператор.	2,25	1	1,25
2.2	Вложенные условные операторы.	2,25	1	1,25
2.3	Условные операторы, логические операции И, ИЛИ, НЕ	2,25	1	1,25
2.4	Оператор выбора. Циклы с предусловиями	2,25	1	1,25
2.5	Циклы с постусловиями	2,25	1	1,25
2.6	Циклы с параметрами	2,25	1	1,25
2.7	Вложенные циклы	2,25	1	1,25
2.8	Решение задач с вложенными циклами	2,25	1	1,25
2.9	Составление программ с ветвлением.	2,25	1	1,25
2.10	Решение задач со строками	2,25	1	1,25
3	Структурированные типы данных	27		
3.1	Одномерные массивы	2,25	1	1,25
3.2	Поиск максимального и минимального значения в массиве	2,25	1	1,25
3.3	Алгоритмы информационного поиска	2,25	1	1,25
3.4	Разбор олимпиадных задач	2,25	1	1,25
3.5	Двумерные массивы	2,25	1	1,25
3.6	Обработка двумерного массива	2,25	1	1,25

3.7	Алгоритмы сортировки. Строковый тип данных	2,25	1	1,25
3.8	Процедуры для работы со строковым типом данных	2,25	1	1,25
3.9	Операции поиска и замены в символьных строках и массивах.	2,25	1	1,25
3.10	Программирование алгоритмов обработки строк.	2,25	1	1,25
3.11	Файловый тип данных	2,25	1	1,25
3.12	Основные операции с файлами	2,25	1	1,25
4	Процедуры и функции	10		
4.1	Рекурсивные функции	2,25	1	1,25
4.2	Общие сведения о подпрограммах. Определение и вызов подпрограмм.	2,25	1	1,25
4.3	Область видимости и время жизни переменной. Механизм передачи параметров.	2,25	1	1,25
4.4	Итоговый проект, решение задачи по выбору учащегося	3,25	2	1,25
	Итого:	64	29	35

IV. СОДЕРЖАНИЕ ПРОГРАММЫ

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 2.5. 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 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.

14

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 программирования и отладка программ

Стил программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

Учащиеся должны знать / понимать:

- что такое стил 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.

V. КВАЛИФИКАЦИОННЫЕ ТРЕБОВАНИЯ К ПЕДАГОГУ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ

Высшее профессиональное образование или среднее профессиональное образование в области, соответствующей профилю кружка, секции, студии, клубного и иного детского объединения без предъявления требований к стажу работы либо высшее профессиональное образование или среднее профессиональное образование и дополнительное профессиональное образование по направлению "Образование и педагогика" без предъявления требований к стажу работы.

VI. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

- Кабинет информатики
- компьютеры или ноутбуки – 10шт.
- столы 10 шт.
- стулья 20 шт.

Программное обеспечение:

- Операционная система: Windows XP (или выше).
- Среда разработки: Python 3.3 (или выше)

VII. МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Список литературы и источников

1. Домашняя страница Python 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 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.

Тест для контроля уровня теоретических знаний по программе

Тест «Язык Python»

Задание: Добавьте каждое предложение по смыслу. За каждый правильный ответ вы набираете 1 балл.

1. Программа Python называется ...
2. Расширение файла Python – as. ...
3. Переменная в Python – это ...
4. Регистр букв в идентификаторах значение ...
5. Выражение в Python – это ...
6. Символ # в Python обозначает ...
7. ... в Python это тип данных для вещественных чисел, встроенный в Python по-умолчанию.
8. Операция $3*4$ - это
9. 345 - ... тип данных.
10. Операция $46\%10$ – это ...
11. Функция `round(d)` – это ...
12. Функция `input()` – предназначена для ...
13. Для вывода данных есть функция в Python - ...
14. ... в Python это логический тип данных, встроенный в Python по-умолчанию.
15. Строки – это ...
16. `A='pri', s='vet'. A+s` – это ...
17. `E='no'. E*5` – это ...
18. К элементу в строке можно обратиться по ...
19. `s='asdfgh'`
`print(s[-1])`. Программа выведет ...
20. `s='asdfgh'`
`print(s[2:4])`. Программа выведет ...
21. Функция `len(строка)` – возвращает ...
22. Списки – это ...
23. Пример списка - ...
24. Словари – это ...
25. Пример словаря - ...
26. Условный оператор в Python - ...
27. Цикл `for` называется циклом ...
28. Переведите конструкцию языка
`S=[1,2,3]`
`for I in s:`
`print(I*4)`
29. Функция `range()` переводится как ...
30. Переведите конструкцию языка
`S=0`
`While S<10:`
`print(S)`
`S=S+1`
31. Функции — это ...

32. Локальные переменные объявлены ...

Ответы:

1. Скрипт
2. Py
3. имя/идентификатор, который может принимать некоторое значение.
4. Имеет
5. это фрагмент языка программирования, представляющий способ вычисления некоторого значения.
6. Комментарий
7. Float
8. Возведение в степень
9. Целочисленный, int
10. Остаток от деления
11. Округление числа
12. Ввода данных в строку
13. Print()
14. Bool
15. Упорядоченные неизменяемые последовательности символов, используемые для хранения и представления текстовой информации
16. Объединение, сложение строк. Конкатенация
17. Повторение строки 5 раз. Дублирование
18. Индексу
19. h
20. dfg
21. длину строки
22. изменяемая последовательность произвольных объектов.
23. C=[2,3,4.5,'gh']
24. Изменяемые неупорядоченные коллекции произвольных объектов с доступом по ключу
25. K={1:'a',2:'b',3:'c'}
26. If
27. Обхода
28. для каждого элемента I в списке s делать следующее (то, что в теле цикла)
29. диапазон
30. пока условие истинно, то выполняется инструкция, после чего условие проверяется снова и снова выполняется инструкция. Так продолжается до тех пор, пока условие будет истинно, в противном случае мы выйдем из цикла.
31. такие участки кода, которые изолированы от остальной программы и выполняются только тогда, когда вызываются.
32. внутри функции.

Ключ:

№ п/п	Количество набранных баллов	Оценка
1	31-32	5
2	27-30	4
3	20-26	3
4	до 19	2

Календарно - тематическое планирование

(занятия проводятся один раз в неделю по 2,25 часа)

	Наименование разделов и тем	Количество часов	Дата		
			Сроки проведения		Корректировка сроков проведения уроков
			1-группа	2-группа	
1.	Знакомство с Python	4,5			
1.1	Техника безопасности. Общие сведения о языке.	2,25			
1.2	Структура программы на языке Python	2,25			
2	Основные алгоритмические конструкции	22,5			
2.1	Условный оператор.	2,25			
2.2	Вложенные условные операторы.	2,25			
2.3	Условные операторы, логические операции И, ИЛИ, НЕ	2,25			
2.4	Оператор выбора. Циклы с условиями	2,25			
2.5	Циклы с постусловиями	2,25			
2.6	Циклы с параметрами	2,25			
2.7	Вложенные циклы	2,25			
2.8	Решение задач с вложенными циклами	2,25			
2.9	Составление программ с ветвлением.	2,25			
2.10	Решение задач со строками	2,25			
3	Структурированные типы данных	27			
3.1	Одномерные массивы	2,25			
3.2	Поиск максимального и минимального значения в массиве	2,25			
3.3	Алгоритмы информационного поиска	2,25			
3.4	Разбор олимпиадных задач	2,25			

3.5	Двумерные массивы	2,25			
3.6	Обработка двумерного массива	2,25			
3.7	Алгоритмы сортировки. Строковый тип данных	2,25			
3.8	Процедуры для работы со строковым типом данных	2,25			
3.9	Операции поиска и замены в символьных строках и массивах.	2,25			
3.10	Программирование алгоритмов обработки строк.	2,25			
3.11	Файловый тип данных	2,25			
3.12	Основные операции с файлами	2,25			
4	Процедуры и функции	10			
4.1	Рекурсивные функции	2,25			
4.2	Общие сведения о подпрограммах. Определение и вызов подпрограмм.	2,25			
4.3	Область видимости и время жизни переменной. Механизм передачи параметров.	2,25			
4.4	Итоговый проект, решение задачи по выбору учащегося	3,25			
	Итого:	64			