

**Пояснительная записка**

**Направленность программы**: дополнительная общеобразовательная общеразвивающая программа «PYTHON» имеет техническую направленность и разработана в соответствии с:

- Федеральным Законом РФ от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;

- Приказом Министерства просвещения РФ от 27 июля 2022 г. N 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

- Концепцией развития дополнительного образования детей до 2030 года (Распоряжение Правительства РФ от 31 марта 2022 г. N 678-р)

- Постановлением Главного государственного санитарного врача России от28.09.2020 № 28 СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

- Приказом Министерства просвещения РФ от 3 сентября 2019 г. № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»;

- Уставом МБОУ Пригорской СШ;

- Письмом Министерства образования и науки Российской Федерации от 18 ноября 2015 г. № 09-3242 «О направлении информации» «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)

- Приказом Департамента Смоленской области по образованию и науке №959-ОД от 18.12.2020 «О реализации мероприятий комплекса мер («дорожной карты») по созданию и функционированию в общеобразовательных организациях, расположенных в сельской местности и малых городах, центров образования естественно-научной и технологической направленностей «Точка роста» в Смоленской области»;

- Методическими рекомендациями по созданию и функционированию региональной сети Центров образования цифрового и гуманитарного профилей «Точка роста» (утверждены 25.06.2020 года);

Программа разработана на основе дополнительной общеобразовательной общеразвивающей программы «Основы программирования на языке Python».

Данная программа и составленное тематическое планирование рассчитано на 1 час в неделю для с 15-16 летнего возраста (1 час). Для реализации программы в кабинете имеется все необходимое оборудование (компьютер, проектор)..

* **Направленность:** дополнительной общеобразовательной общеразвивающей программы–*техническая.*

# Новизна программы

Новизна Программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

**Актуальность:** с развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью научиться программировать.

Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями.

Языки программирования можно сравнить с иностранными языками, овладеть ими может каждый. Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

**Срок реализации программы:** 1 год обучения (учебных недель 36).

**Формы организации образовательного процесса**

* групповая,
* фронтальная,
* индивидуальная,
* парная.

**Виды занятий:**

* теоретические
* практические
* мастер-класс
* соревнование,
* экскурсия,
* конкурсы

Программа может корректироваться с учетом имеющейся материально-технической базы и контингента обучающихся. Количество детей в группе - 10 человек.

**Материально-техническое обеспечение**

**Компьютерное оборудование:**

* Персональные компьютеры для работы с предустановленной операционной системой и специализированным ПО

**Профильное оборудование:**

* Компьютер (системный блок, монитор, мышь, клавиатура)с доступом к сети Интернет);
* Мультимедийный проектор;
* Доска магнитно-маркерная.

**Программное обеспечение:**

* Операционная система;
* Антивирусная программа;
* Офисные приложения;
* Интернет-браузеры последней версии;
* Компилятор Python 3.5.

**Дополнительное оборудование:**

* БумагаА4.
* Маркеры для магнитно-маркернойдоски.
* Губка для магнитно-маркернойдоски.

**Методическая работа**

1. Составление методических разработок

2. Мониторинг образовательного процесса

З. Посещение занятий педагогов других объединений с целью обмена опыта. Ожидаемые результаты и способы их проверки:

Формы подведения итогов реализации ДОП:

школьная, муниципальная, региональная состязания по прогаммированию;  турниры на звание лучшего программиста. Данная программа направлена на:

 помощь детям в индивидуальном развитии; - мотивацию к познанию и творчеству:  к стимулированию творческой активности;  развитию способностей к самообразованию;  приобщение к общечеловеческим ценностям;  организацию детей во внеучебное время (досуг). На занятиях создана структура деятельности, создающая условия для творческого развития воспитанников на различных возрастных этапах и предусматривающая их дифференциацию по степени одаренности. Основные дидактические принципы программы: доступность и наглядность, ориентация на результат. Такую стратегию обучения помогает реализовать образовательная среда.

Эффективность обучения основам программирования зависит и от организации занятий проводимых с применением следующих методов:

* Объяснительно иллюстративный предъявление информации различными способами (объяснение, рассказ, беседа, инструктаж, демонстрация, работа с технологическими картами );
* Эвристический - метод творческой деятельности (создание творческих моделей)
* Проблемный постановка проблемы и самостоятельный поиск её решения обучающимися;
* Программированный - набор операций, которые необходимо выполнить в ходе выполнения практических работ (форма: компьютерный практикум, проектная деятельность);
* Репродуктивный воспроизводство знаний и способов деятельности (форма: собирание моделей и конструкций по образцу, беседа, упражнения по аналогу),  Частично - поисковый - решение проблемных задач с помощью педагога;
* Поисковый — самостоятельное решение проблем;
* Метод проблемного изложения - постановка проблемы педагогам, решение ее самим педагогом, соучастие обучающихся при решении. И все-таки, главный метод - это метод проектов как технология организации образовательных ситуаций, в которых учащийся ставит и решает собственные задачи, и технология сопровождения самостоятельной деятельности учащегося
* Проектно-ориентированное обучение - это систематический учебный метод, вовлекающий учащихся в процесс приобретения знаний и умений с помощью широкой исследовательской деятельности, базирующейся на комплексных, реальных вопросах и тщательно проработанных заданиях.

**Цель программы:** обучение учащихся программированию посредством языка Python, развитие инженерного мышления, воспитание конкурентно способной личности.

**Задачи**:

## Обучающие:

* обучить языку программирования Puthon и созданию программ на его основе;
* научить создавать прикладное программное обеспечение;
* расширять кругозор обучающихся в области программирования;
* научить дизайнерскому оформлению созданного ПО.

## Развивающие:

* развивать память и внимание, познавательную и творческую активность;
* развивать творческие способности, эстетическое и эргономическое восприятие объектов труда;
* развивать логическое мышление.

## Воспитательные:

* прививать интерес к активному творческому самовыражению, культуре труда;
* воспитывать упорство в достижении желаемого результата;
* воспитывать эстетический вкус;
* воспитывать чувство взаимопомощи, доверия, коллективизма.

**Планируемые результаты:**

**Предметные результаты:**

В результате освоения программы обучающиеся должны

*знать*:

* основные алгоритмические конструкции;
* принципы построения блок-схем;
* принципы структурного программирования на языке Python;

*уметь*:

* составлять алгоритмы для решения прикладных задач;
* реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
* применять библиотеку Tkinter;
* отлаживать и тестировать программы, написанные на языке Python;
* представлять свой проект.

*владеть*:

* основной терминологией в области алгоритмизации и программирования;
* основными навыками программирования на языке Python;

**Личностные результаты:**

* критическое отношение к информации и избирательность её восприятия;
* осмысление мотивов своих действий при выполнении заданий;
* развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
* развитие внимательности, настойчивости, целеустремлённости, умения преодолевать трудности;
* развитие самостоятельности суждений, независимости и нестандартности мышления;
* освоение социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах;
* формирование коммуникативной компетентности в общении и сотрудничестве с другими обучающимися.

**Метапредметные результаты:**

* ориентироваться в своей системе знаний: отличать новое знание от известного;
* перерабатывать полученную информацию: делать выводы в результате совместной работы группы, сравнивать и группировать предметы и их образы;
* работать по предложенным инструкциям исамостоятельно;
* излагать мысли в четкой логической последовательности, отстаивать свою точку зрения, анализировать ситуацию и самостоятельно находить ответы на вопросы путём логических рассуждений;
* определять и формировать цель деятельности на занятии с помощью учителя;
* работать в группе и коллективе;
* уметь рассказывать о проекте;
* работать над проектом в команде, эффективно распределять обязанности.

**Учебно-методическое и информационное обеспечение Программы:**

1. Гин, А.А. Приёмы педагогической техники: свобода выбора, открытость, деятельность, обратная связь, идеальность: По- собие для учителей / А.А. Гин. — Гомель: ИПП «Сож», 1999 — 88 с.
2. Бреннан, К. Креативное программирование / К. Бреннан, К. Болкх, М. Чунг. — Гарвардская Высшая школа образования, 2017
3. Лутц, М. Программирование на Python. Т. 1 / М. Лутц. — М.:

Символ, 2016 — 992 c.

1. Лутц, М. Программирование на Python. Т. 2 / М. Лутц. — М.: Символ, 2016 — 992 c.
2. Бриггс, Джейсон. Python для детей. Самоучитель по программированию / Джейсон Бриггс. — МИФ. Детство, 2018 — 320 с.

**Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» поддерживает образовательный процесс в организациях, реализующих Концепцию развития дополнительного образования в условиях Центров образования цифрового и гуманитарного профилей «Точка роста».**

**2. Учебно-тематический план**

**Учебный план**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *№п/п* | *Название раздела, темы* | *Количество часов* | | |
| *Всего* | *Теория* | *Практика* |
| 1. | Знакомство с языком Python | 4 | 2 | **2** |
| 2 | Переменные и выражения | 7 | 3 | 4 |
| 3 | Условные операторы | 6 | 2 | 4 |
| 4 | Циклы | 9 | 3 | 6 |
| 5 | Функции | 5 | 2 | 3 |
| 6 | Строки | 4 | 2 | 2 |
| 7 | Итоговое тестирование по курсу | 1 |  | 1 |
|  |  | 36 | 14 | 22 |

**Содержание программы**

**Тема 1. Знакомство с языком Python (4 ч.)**

Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программ на языкеPython. Комментарии.

 Практическая работа 1.1. Установка программы Python.

 Практическая работа 1.2. Режимы работы с 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 интерактив- ной среде;написать комментарии в программе.

**Тема 2. Переменные и выражения (7 ч.)**

Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения

операций. Математическиефункции.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 элементар- ные действия с числами.

 Практическая работа 2.1. Переменные

 Практическая работа 2.2. Выражения

 Практическая работа 2.3. Задачи на элементарные действия с числами Самостоятельная работа 1 «Решение задач на действия с числами».

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 выра- жения.

**Тема 3. Условные операторы 6 ч.)**

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Примеры реше- ния 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

Самостоятельная работа 2. Решение задач по теме "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 (9 ч.)**

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с усло- вием. 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 3 по теме "Циклы"

Учащиеся должны знать / понимать:

циклы с условием и их виды; правила записи циклов условием; назначение и особен- ности использования цикла с параметром; формат записи цикла с параметром; при- меры использования циклов различных типов.

Учащиеся должны уметь:

определять вид цикла, наиболее удобный для решения поставленной задачи;

использовать цикл с условием; определять целесообразность применения и использо- вать цикл с параметром для решения поставленной задачи;

**Тема 5. Функции (5 ч.)**

Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, ин- струкция lambda. Примеры решения задач c использованием функций. Рекурсивны функции. Вычисление факториала. Числа Фибоначчи.

 Практическая работа 5.1. Создание функций

 Практическая работа 5.2. Решение задач с использованием функций

 Практическая работа 5.3. Рекурсивные функции

Учащиеся должны знать / понимать:

понятие функции;способы описания функции;принципы структурного программиро- вания;понятие локальных переменных подпрограмм;понятие формальных и фактиче- ских параметров подпрограмм;способ передачи параметров.

Учащиеся должны уметь:

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

**Тема 6. Строки (4 ч.)**

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

Срезы строк. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строка- ми.

 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. Итоговое тестирование по курсу** (1 ч.)

# 3. Формы аттестации и оценочные материалы

В процессе реализации программы предусмотрены следующие виды контроля:

* ***входной контроль проводится*** с целью определения уровня знаний

учащихся (Приложение № 1);

* ***промежуточный контроль*** проводится регулярно на занятиях с целью определения степени усвоения материала в форме опроса, решения задач и практических заданий;
* ***итоговый контроль*** – защита проекта.

# Формы подведения итогов реализации программы

Результаты обучения по программе выявляются по итогам проведения олимпиад, соревнований по программированию, защиты проекта.

**4. Календарно-тематическое планирование**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № п/п | | Дата | | Кол-во часов | Тема урока |
| план | факт |
| **Тема 1.Знакомство с языком Python** | | | | | |
|  |  | |  | 1 | Общие сведения о языке Python. |
|  |  | |  | 1 | Практическая работа 1.1. Установка программы Python. |
|  |  | |  | 1 | Что такое программа. Структура программ на языке Python. |
|  |  | |  | 1 | Практическая работа 1.2. Режимы работы с Python |
| **Тема 2.Переменные и выражения** | | | | | |
|  |  | |  | 1 | Переменные. Практическая работа 2.1. Переменные |
|  |  | |  | 1 | Выражения. Ввод и вывод |
|  |  | |  | 1 | Практическая работа 2.2. Выражения |
|  |  | |  | 1 | Задачи на элементарные действия с числами действия с числами |
|  |  | |  | 1 | Практическая работа 2.3. Задачи на элементарные дей- ствия с числами. |
|  |  | |  | 1 | *Самостоятельная работа 1 «*Решение задач на действия с числами». |
| **Тема 3. Условные операторы** | | | | | |
|  |  | |  | 1 | Логические выражения и операторы |
|  |  | |  | 1 | Практическая работа 3.1. Логические выражения |
|  |  | |  | 1 | Условный оператор. Множественное ветвление. |
|  |  | |  | 1 | Условный оператор. Множественное ветвление. |
|  |  | |  | 1 | Условный оператор. Множественное ветвление. |
|  |  | |  | 1 | Практическая работа 3.2. "Условный оператор" |
|  |  | |  | 1 | Практическая работа 3.3. Множественное ветвление |
|  |  | |  |  | *Самостоятельная работа 2*. Решение задач по теме "Условные операторы". |
| Тема 4. Циклы | | | | | |
|  |  | |  | 1 | Оператор цикла с условием. |
|  |  | |  | 1 | Практическая работа 4.1. "Числа Фибоначчи" |
|  |  | |  | 1 | Оператор цикла for. |
|  |  | |  | 1 | Практическая работа 4.2. Решение задачи с циклом for. |
|  |  | |  | 1 | Вложенные циклы. Случайные числа. |
|  |  | |  | 1 | Практическая работа 4.3. Реализация циклических алго- ритмов. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | 1 | Практическая работа 4.4. Случайные числа. |
|  |  |  | 1 | Практическая работа 4.5. Решение задач с циклом |
|  |  |  | 1 | *Самостоятельная работа 3 по теме "Циклы".* |
| **Тема 5. Функции** | | | | |
|  |  |  | 1 | Создание функций. Локальные переменные |
|  |  |  | 1 | Практическая работа 5.1. Создание функций |
|  |  |  | 1 | Практическая работа 5.2. Решение задач с использовани-  ем функций. Рекурсивные функции |
|  |  |  | 1 | Практическая работа 5.3. Рекурсивные функции |
| **Тема 6. Строки** | | | | |
|  |  |  | 1 | Строки |
|  |  |  | 1 | Практическая работа 6.1. Строки |
|  |  |  | 1 | Срезы строк |
|  |  |  | 1 | Практическая работа 6.2. Решение задач со строками. |
| **Тема 7. Промежуточная аттестация.** | | | | |
|  |  |  | 1 | **Промежуточная аттестация. Тест.** |

# Список литературы

1. Доусен М. Программируем на Python / М. Доусен - СПб.: Питер, 2016. - 416c.
2. Лутц М. Изучаем Python, 4 издание / М. Лутц - СПб.: Символ- Плюс, 2011. - 1280 с.
3. Любанович Б. Простой Python. Современный стиль программирования / Б. Любанович. - СПб.: Питер, 2016. - 480c.
4. Прохоренок H.А., Дронов В.А. Python 3 и PyQt 5. Разработка приложений / H.А. Прохоренок, В.А. Дронов - СПб.: «БХВ- Петербург», 2016. - 832с.
5. Саммерфильд М. Python на практике / М. Саммерфильд, пер. А.А. Слинкин – М.: ДМК-Пресс, 2014. - 338с.

# Цифровые образовательные ресурсы

1. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С., Задка М., Левис М., Монтаро С., Реймонд Э.С., Кучлинг А.М., Лембург М.-А., Йи К.-П., Ксиллаг Д., Петрилли Х.Г., Варсав Б.А., Ахлстром Дж.К., Роскинд Дж., Шеменор Н., Мулендер С. Язык программирования Python: [Электронный ресурс]. 2001. URL: [https://goo.gl/8TzY8w.](https://goo.gl/8TzY8w) (Дата обращения: 27.08.2018).

**Требования к комплектации компьютерного класса**

 12–15 компьютеров (рабочих мест) для обучающихся и один компьютер (рабо- чего места) для педагога.

 компьютеры объединены в локальную сеть с возможностью выхода в Интер- нет.

Минимальные требования к техническим характеристикам каждого компью- тера следующие:

 процессор – с тактовой частотой 2 ГГц;

 оперативная память – не менее 1 Гб;

 жидкокристаллический монитор с диагональю не менее 15 дюймов;

 жёсткий диск – не менее 250 Гб;

 клавиатура;

 мышь;

 аудио-карта и акустическая система (наушники или колонки).

 Мультимедийный проектор на рабочем месте учителя;

Требования к программному обеспечению компьютеров:

 операционная система Windows или Linux

 текстовый редактор (Блокнот) и текстовый процессор (MS Word или OpenOffice Writer);

 табличный процессор (MS Excel или OpenOffice Calc);

 среда программирования Python 3+.

# Приложение № 1

1. Запишите значение переменной *t*, полученное в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

|  |  |  |
| --- | --- | --- |
| Алгоритмический язык | Бейсик | Паскаль |
| алг  нач цел t,i t := 2  нц для i от 1 до 3 t :=t \* i кц  вывод t кон | DIM i,t AS INTEGER  t = 2  FOR i = 1 TO 3  t = t \* i NEXT i PRINT t | Vart,i: integer;  Begin t := 2;  For i := 1 to 3 do t :=t \* i; Writeln(t); End. |

Ответ:

1. В таблице Dat хранятся данные о количестве краткосрочных командировок, в которые приходилось ездить сотруднику за последний год (Dat[1] — количество командировок в январе, Dat[2] — количество командировок в феврале и т. д.). Определите, что будет напечатано в результате выполнения следующего алгоритма, записанного на трёх алгоритмических языках.

|  |  |  |
| --- | --- | --- |
| Алгоритмический язык | Бейсик | Паскаль |
| алг нач  целтаб Dat[ 1:12] цел к, ш, month Dat[l] := 5 Dat[2] := 5 Dat[3] := 6 Dat[4] := 8 Dat[5] :=  4 Dat[6] := 5 Dat[7] := 4 Dat[8]  := 7 Dat[9] := 4 Dat[10] :=4  Dat[ll] := 8 Dat[12] := 7 month:=l; m := Dat[l] нц для к от 2 до 12 если Dat[K] <= m то m:= Dat[k]; month := k все КЦ вывод month КОН | DIM Dat(12) AS INTEGER  Dat[l] = 5  Dat[2] = 5  Dat[3] = 6  Dat[4] = 8  Dat[5] = 4  Dat[6] = 5  Dat[7] = 4  Dat[8] = 7  Dat[9] := 4  Dat[10] :=4  Dat[ll] := 8  Dat[12] := 7  month = 1: m = Dat(l) FORk = 2 TO 12  IF Dat(k) <= m THEN  m = Dat(k) month = k END IF NEXT k  PRINT month END | Var k, m, month: integer; Dat: array[1. .12] of integer; Begin  Dat[l] := 5;  Dat[2] := 5;  Dat[3] := 6;  Dat[4] := 8;  Dat[5] := 4;  Dat[6] := 5;  Dat[7] := 4;  Dat[8] := 7;  Dat[9] := 4;  Dat[10] :=4;  Dat[ll] := 8;  Dat[12] := 7; month: = 1; m := Dat[l]; for k := 2 to 12 do if Dat[k] <= m then begin  m:= Dat[k]; month := k; end;  write(month); End. |

Ответ:

1. У исполнителя Квадратор две команды, которым присвоены номера: 1. возведи в квадрат, 2. прибавь 3.

Первая из них возводит число на экране во вторую степень, вторая — прибавляет к числу 3. Составьте алгоритм получения из числа 5 числа 127, содержащий не более 5 команд. В ответе запишите только номера команд. *(Например, 12212* — *это алгоритм: возведи в квадрат прибавь 3 прибавь 3 возведи в квадрат прибавь 3,*

*который преобразует число 2 в число 103).*

Если таких алгоритмов более одного, то запишите любой из них. Ответ:

1. Некоторый алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала вычисляется длина исходной цепочки символов; если она чётна, то в середину цепочки символов добавляется символ А, а если нечётна, то в начало цепочки добавляется символ Б. В полученной цепочке символов каждая буква заменяется буквой, следующей за ней в русском алфавите (А — на Б, Б — на В и т. д., а Я — на А). Получившаяся таким образом цепочка является результатом работы алгоритма.

Например, если исходной была цепочка **ВРМ,** то результатом работы алгоритма будет цепочка **ВГСН,** а если исходной была цепочка **ПД,** то результатом работы алгоритма будет цепочка **РБЕ.**

Дана цепочка символов **ТОР.** Какая цепочка символов получится, если к данной цепочке применить описанный алгоритм дважды (т. е. применить алгоритм к данной цепочке, а затем к результату вновь применить алгоритм)? Русский алфавит: АБВГДЕЁЖЗИИКЛМНОПРСТУ- ФХЦЧШЩЪЫЬЭЮЯ.

Ответ:

1. Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 6. Количество чисел не превышает 1000. Введённые числа не превышают 30 ООО. Программа должна вывести одно число — количество чисел, оканчивающихся на 6.

# Пример работы программы:

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 3, 16, 26, 24 | o, z |