

**Муниципальное бюджетное учреждение
дополнительного образования
«Центр внешкольной работы»
Бузулукского района
Оренбургской области**

«Рассмотрено»
на заседании методического совета
МБУ ДО «Центр внешкольной работы»
Бузулукского района
«16» сентября 2020 г.

«Утверждаю»
Директор МБУ ДО
«Центр внешкольной работы»
Е.Н. Филатова
«16» сентября 2020 г.



**Дополнительная общеобразовательная общеразвивающая программа
технической направленности
ОЗШ «РОСТ» (информатика)**

Возраст обучающихся: 13-15 лет
срок реализации программы: 1 год

Составитель:
Дивеева Светлана Александровна,
педагог дополнительного образования

п. Красногвардеец
2020 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Направленность **дополнительной** общеобразовательной общеразвивающей программы ОЗШ «РОСТ» (информатика) – **техническая**.

Программа построена таким образом, чтобы помочь обучающимся заинтересоваться информационными технологиями вообще и найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни при работе с большим объемом информации. Компьютер не заменяет традиционное занятие, а только дополняет его.

Работа с графикой на компьютере всё больше и больше становится неотъемлемой частью компьютерной грамотности любого человека. Люди самых разных профессий применяют компьютерную графику в своей работе.

Для обучающихся данного возраста – это понимание того, для чего нужен компьютер, в каких сферах жизни он используется, сколько можно работать по времени, как правильно обращаться с техникой. Если эти задачи выполняются, то в дальнейшем обучающийся детского объединения не будет воспринимать компьютер только как приставку для игр, а как многогранное устройство с бесконечными возможностями для образования, для творчества.

Знания и умения, приобретенные в результате освоения программы, являются фундаментом для дальнейшего совершенствования мастерства в области моделирования, анимации, видеомонтажа.

Новизна дополнительной образовательной программы предполагает изучать углубленно темы по Программированию на языке Python.

А также включено изучение методов построения анимационных изображений, ретуширования и восстановления изображений, а также компьютерные программы, с которыми обучающиеся работают:

1. Компьютерный тренажер;
2. Компьютерная программа Кирилла и Мефодия «Мир информатики»;
3. Программа «Мульти-Пульти» для создания мультфильмов;
4. Программа Adobe PhotoshopCS5;
5. Компьютерная программа для рисования «Tux Paint»;
6. Программа для создания видеороликов «Movie Maker»;
7. Компьютерная программа для начинающих «Miniseite».

Актуальность программы в том, что программа позволяет педагогу концентрировать внимание на индивидуальности каждого обучающегося, помогает развитию личности через техническое творчество. Занятия в объединении «Мир информатики» стимулируют творческую деятельность, создают условия для развития личностных качеств обучающихся.

Актуальность программы обусловлена тем, что в современное время дети учатся по развивающим технологиям, где логическое мышление является основой.

Программа определяется по запросам со стороны детей и их родителей на изучение компьютерной грамотности.

Педагогическая целесообразность

Данная образовательная программа педагогически целесообразна, т.к. при ее реализации обучающийся сможет сам сформулировать задачи, новые знания теории помогут ему в процессе решения творческих заданий.

На занятиях позволяет сохранить высокую творческую работоспособность обучающихся. Так же обучающиеся создают индивидуальные проекты.

Цель: научиться программировать на языке Python, развитие поэтапного логического мышления, алгоритмического мышления обучающихся, наблюдательности. Сформировать информационную культуру личности обучающихся, способных к творческому самовыражению через овладение компьютерных программ.

Задачи:

образовательные

- расширение представления обучающихся о возможностях компьютера, областях его применения;
- углубление обучающихся в мире современных профессий,
- знакомство на практике с деятельностью художника;
- обучение навыкам самостоятельного использования компьютера в качестве средства для решения практических задач.

развивающие

- Развивать внимание, память, логическое и пространственное воображения.
- развитие трудовой и творческой активности;
- Развитие умения ориентироваться в интернет - пространстве;
- Развитие памяти, внимания, воображения и фантазии;
- Развитие умений гармонично сочетать цвета;
- Развитие умений создавать композицию.

воспитательные

- формирование творческого подхода к поставленной задаче;
- формирование установки на позитивную социальную деятельность в информационном обществе, на недопустимость действий, нарушающих правовые и этические нормы работы с информацией;
- ознакомление с православной культурой, в ее знаково-символических проявлениях; в том числе, в праздниках.
- воспитание высоких духовно-нравственных качеств: милосердия, доброты, отзывчивости, уважительного отношения к старшим, честности, искренности;
- Формирование и развитие дисциплинированности, добросовестного отношения к занятиям.
- формирование умения правильно определять жизненные приоритеты, ориентируясь на традиционные духовно-нравственные ценности.

Отличительные особенности, данная программа позволяет индивидуализировать сложные проекты: более сильным обучающимся будет интересна сложная конструкция, менее подготовленным, предлагается работа проще. При этом обучающий и развивающий смысл работы сохраняется. Это дает возможность предостеречь обучающегося от страха перед трудностями, приобщить без боязни творить и создавать.

Использование данной программы дает возможность каждому обучающемуся попробовать свои силы в разных видах технического творчества, выбрать приоритетное направление и максимально реализовать себя в нем.

Возраст обучающихся, участвующих в реализации данной образовательной программы 13-15 лет.

Ознакомление с программным обеспечением в области обучающих, развивающих программ, самостоятельную постановку задачи, структурирование и преобразование информации в текстовую и мультимедийную форму. Общение с компьютером увеличивает потребность в приобретении знаний.

Условия набора обучающихся. В объединение зачисляются все, желающие обучающиеся, без предварительного отбора по заявлению родителей.

Формы и режим занятий

Занятия проводятся в разнообразных формах, с использованием различных методов:

- используется словесный метод: беседа, рассказ, обсуждение, игра;
- практическая работа является основной формой проведения занятия;
- используется наглядный метод – демонстрация иллюстраций, презентаций, образцов изделий и т.п.
- творческая работа;
- создание самостоятельных творческих работ и работ по шаблону на основе теоретических знаний;
- подготовка к выставке.
- иллюстративный (объяснение сопровождается демонстрацией наглядного материала),
- репродуктивный (воспроизведение).

В проведении занятий используются групповые, индивидуальные и коллективные формы работы:

- групповая (используется на практических занятиях, в самостоятельной работе обучающихся, в подготовке к конкурсам и т.д.);
- индивидуальная (используется при подготовке и выполнении творческих работ, индивидуальных выставок, проектов);
- коллективная (используется на общих занятиях).

Формы занятий: традиционное занятие, комбинированное занятие; лекция, практическое занятие, игра, праздник, путешествие, экскурсия, соревнование, выставка.

Теоретическая часть даётся в форме бесед с просмотром иллюстративного и наглядного материалов и закрепляется практическим освоением темы.

В программе отводятся часы на индивидуальные занятия, которые необходимы для подготовки к конкурсам и мероприятиям или для занятий с обучающимися, которые по каким-либо причинам отстали от программы, а также с высокомотивированными детьми.

Общий объем времени по плану определен количеством рабочих недель в учебном году: 36 недель.

Формы подведения итогов

Промежуточным и конечным итогом работы является занятие, где обучающиеся выполняют индивидуальную, творческую работу;

- организация и проведение тематических выставок;
- мастер-класс;
- открытое занятие.

Праздничные мероприятия, игры – это своего рода контроль среза знаний, умений и навыков, полученных на занятиях. Организация выставок – это контроль роста обучающегося, способ выражения творчества, воспитание ответственности и желания работать интереснее.

Формы подведения итогов по реализации образовательной программы

- участие в конкурсах различного уровня;
- открытые занятия;
- итоговые (аттестационные) занятия;
- показ детских достижений на тематических выставках, конференциях и т.д.

Способы определения результативности образовательного процесса используются следующие виды контроля:

начальный контроль (сентябрь)

текущий контроль (в течение года)

итоговый контроль (май)

Ожидаемые результаты

Первый уровень

В результате изучения курса на первом уровне учащийся

- 1) научится составлять и отлаживать простые диалоговые программы;
- 2) узнает особенностей машинных вычислений с целыми и вещественными числами;
- 3) научится использовать основные алгоритмические конструкции: условные операторы, циклы с условием, циклы по переменной;
- 4) овладеет методами построения графических изображений программными средствами;

5) овладеет простыми методами программирования компьютерной анимации.

Второй уровень

В результате изучения курса на втором уровне учащийся

- 6) познакомится с методами проектирования программ «сверху вниз» и «снизу вверх»;
- 7) научится использовать вспомогательные алгоритмы (процедуры и функции) для структуризации программ;
- 8) научится применять рефакторинг для улучшения читаемости программ;
- 9) научится использовать символьные строки;
- 10) овладеет основными алгоритмами обработки одномерных и двумерных массивов;
- 11) познакомится с понятием сложности алгоритма;

Третий уровень

В результате изучения курса на третьем уровне учащийся

- 12) научится применять различные алгоритмы сортировки массивов;
- 13) научится использовать двоичный поиск;
- 14) научится обрабатывать данные, записанные в текстовые и двоичные файлы, и сохранять в файлах результаты работы программы;
- 15) научится использовать структуры для объединения данных;
- 16) научится применять словари, стеки, очереди, деки для решения задач обработки данных;
- 17) научится использовать деревья для организации данных;
- 18) познакомится с методами описания графов и некоторыми популярными алгоритмами на графах;
- 19) научится использовать динамическое программирование для решения комбинаторных и оптимизационных задач;
- 20) познакомится с понятием выигрышных и проигрышных позиций в играх с полной информацией;

Четвёртый уровень

В результате изучения курса на четвёртом уровне учащийся

- 21) познакомится с объектно-ориентированным подходом к разработке программ;
- 22) научится выполнять объектно-ориентированный анализ задачи, выделять свойства и методы объектов;
- 23) научится использовать инкапсуляцию для защиты данных объектов;
- 24) познакомится с понятиями «класс» и «абстрактный класс»;

- 25) познакомится с понятиями «инкапсуляция», «наследование», «полиморфизм»;
- 26) научится проектировать несложные иерархии классов для прикладных задач;
- 27) познакомится с принципами разработки событийно-ориентированных программ;
- 28) научится создавать программы с графическим интерфейсом на языках Python и C#;
- 29) научится использовать готовые и создавать новые компоненты (виджеты) для сред быстрой разработки программ.

Содержание программы

Первый уровень

Программирование на языке Python (17 часов)

Простейшие программы. Диалоговые программы. Переменные. Консольный ввод и вывод данных.

Компьютерная графика. Система координат. Управление пикселями. Графические примитивы: линии, прямоугольники, окружности. Изменение координат. Анимация.

Процедуры. Процедуры с параметрами. Рефакторинг.

Обработка целых чисел. Арифметические выражения. Деление нацело. Остаток от деления.

Обработка вещественных чисел. Особенности представления вещественных чисел в памяти компьютера. Операции с вещественными числами.

Случайные и псевдослучайные числа. Генераторы случайных чисел.

Ветвления. Условный оператор. Полная и неполная формы условного оператора. Вложенные условные операторы. Логические переменные. Экспертные системы. Сложные условия. Логические операции И, ИЛИ, НЕ. Порядок выполнения операций.

Циклы с условием. Алгоритм Евклида. Обработка потока данных. Бесконечные циклы. Циклы по переменной. Шаг изменения переменной цикла.

Циклы в компьютерной графике. Узоры. Вложенные циклы. Штриховка.

Второй уровень

Программирование на языке Python (17 часов).

Этапы создания программ. Методы проектирования программ «сверху вниз» и «снизу вверх». Интерфейс и реализация. Документирование программы.

Подпрограммы: процедуры и функции. Процедуры. Процедуры с параметрами. Локальные и глобальные переменные.

Функции. Логические функции.

Рекурсия. Рекурсивные процедуры и функции. Фракталы.

Символьные строки. Сравнение строк. Операции со строками. Обращение к символам. Перебор всех символов. Срезы. Удаление и вставка. Встроенные методы. Поиск в символьных строках. Замена символов. Преобразования «строка — число». Символьные строки в функциях. Рекурсивный перебор.

Массивы (списки). Массивы в языке Python. Создание массива. Обращение к элементу массива. Перебор элементов массива. Генераторы. Вывод массива. Ввод массива с клавиатуры. Заполнение массива случайными числами.

Алгоритмы обработки массивов. Сумма элементов массива. Подсчёт элементов массива, удовлетворяющих условию. Особенности копирования списков в Python.

Поиск в массивах. Линейный поиск. Поиск максимального элемента в массиве. Максимальный элемент, удовлетворяющий условию. Использование массивов в прикладных задачах.

Матриц. Создание и заполнение матриц. Вывод матрицы на экран. Перебор элементов матрицы. Квадратные матрицы.

Сложность алгоритмов. Асимптотическая сложность.

Третий уровень

Программирование на языке Python (18 часов).

Сортировка массивов. Метод пузырька (сортировка обменами). Метод выбора. Сортировка слиянием. Быстрая сортировка. Сортировка в языке Python.

Двоичный поиск в массиве данных. Двоичный поиск по ответу.

Обработка файлов. Типы файлов. Чтение данных. Запись данных. Обработка данных из файла.

Целочисленные алгоритмы. Решето Эратосфена. Целочисленный квадратный корень.

Словари. Алфавитно-частотный словарь. Перебор элементов словаря.

Структуры. Классы. Создание структур. Работа с полями структур. Хранение структур в файлах. Сортировка структур.

Стек. Использование списка. Вычисление арифметических выражений. Скобочные выражения. Системный стек. Очередь. Дек.

Деревья. Деревья поиска. Обход дерева. Использование связанных структур. Вычисление арифметических выражений.

Графы. Описание графа. Жадные алгоритмы. Минимальное остовное дерево. Алгоритм Дейкстры. Алгоритм Флойда–Уоршелла. Использование списков смежности.

Динамическое программирование. Числа Фибоначчи. Количество программ для исполнителя. Двумерные задачи. Поиск оптимального решения.

Игровые модели. Выигрышные и проигрышные позиции.

Четвёртый уровень

Программирование на языке Python (18 часов). Проблема сложности программ. Процедурный и объектно-ориентированный подходы к написанию программ.

Классы и объекты. Объектно-ориентированный анализ. Взаимодействие объектов. Свойства и методы.

Классы и объекты в программе. Объявление класса. Поля класса. Конструктор класса. Данные и методы класса.

Скрытие внутреннего устройства. Доступ к полям через методы. Свойства (*property*). Свойство «только для чтения»

Иерархия классов. Наследование. Базовый класс. Доступ к полям. Классы-наследники. Полиморфизм. Разработка модулей.

Событийно-ориентированное программирование. Программы с графическим интерфейсом. Форма. Свойства формы. Обработчики событий.

Использование компонентов (виджетов). Ввод и вывод данных. Обработка ошибок с помощью исключений.

Создание компонентов. Добавление свойств и методов. Составные компоненты.

Модель и представление.

Тематическое планирование курса

1 уровень (17 часов)

Номер занятия	Тема занятия
	Программирование на языке Python
1.	Первые программы
2.	Диалоговые программы
3.	Компьютерная графика
4.	Процедуры
5.	Обработка целых чисел
6.	Обработка вещественных чисел
7.	Случайные и псевдослучайные числа
8.	Ветвления
9.	Сложные условия
10.	Циклы с условием
11.	Циклы с условием: практикум
12.	Анимация
13.	Циклы по переменной
14.	Циклы в компьютерной графике
15.	Выполнение проекта
16.	Выполнение проекта
17.	Выполнение проекта

2 уровень (17 часов)

Номер занятия	Тема занятия
	Программирование на языке Python
1.	Проектирование программ
2.	Процедуры
3.	Рекурсия
4.	Функции
5.	Символьные строки
6.	Обработка символьных строк
7.	Строки в функциях
8.	Массивы
9.	Ввод и вывод массивов
10.	Суммирование элементов массива
11.	Подсчёт элементов массива, удовлетворяющих условию
12.	Поиск значения в массиве
13.	Поиск максимального элемента в массиве
14.	Игра «Стрельба по тарелкам»
15.	Игра «Стрельба по тарелкам»
16.	Матрицы
17.	Сложность алгоритмов

3 уровень (18 часов)

Номер занятия	Тема занятия
	Программирование на языке Python
1.	Простые алгоритмы сортировки
2.	Сортировка слиянием

Номер занятия	Тема занятия
3.	Быстрая сортировка
4.	Двоичный поиск
5.	Обработка файлов
6.	Обработка файлов: практикум
7.	Целочисленные алгоритмы
8.	Словари
9.	Структуры
10.	Структуры: практикум
11.	Стек, очередь, дек
12.	Деревья
13.	Графы
14.	Графы: практикум
15.	Динамическое программирование
16.	Динамическое программирование: практикум
17.	Игровые модели
18.	Игровые модели: практикум

4 уровень (16 часов)

Номер урока	Тема занятия
	Программирование на языке Python
1.	Что такое ООП?
2.	Модель задачи: классы и объекты
3.	Классы и объекты в программе
4.	Классы и объекты в программе: практикум
5.	Скрытие внутреннего устройства
6.	Иерархия классов
7.	Классы-наследники (I)
8.	Классы-наследники (II)
9.	Доработка игры
10.	Событийно-ориентированное программирование
11.	Использование компонентов (виджетов)
12.	Использование компонентов (виджетов)
13.	Создание компонентов
14.	Модель и представление
15.	Выполнение проекта
16.	Выполнение проекта

Оценивание результатов обучения

Результатом обучения считается способность учащегося написать программу (разработать проект) определённого уровня сложности. Далее выделяются следующие уровни сложности:

- А:** начальный уровень, воспроизведение изучаемого материала с незначительными изменениями;
- В:** средний уровень, способно применять изученный материал для написания программ, которые отличаются от изученных;
- С:** высокий уровень, способно применять изученный материал для самостоятельного написания программ, решающих нестандартные задачи.

Первый уровень

Программирование на языке Python

§ 1. Первые программы

§ 2. Диалоговые программы

А: Напишите программу, которая вводит три целых числа и находит их сумму.

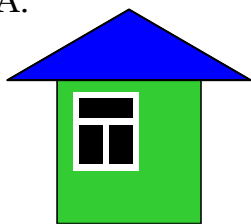
В: Напишите программу, которая вводит три целых числа и находит их сумму и произведение.

С: Напишите программу, которая вводит три целых числа и находит их сумму, произведение и среднее арифметическое.

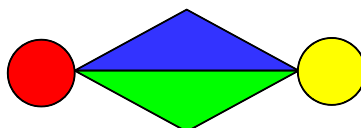
§ 3. Компьютерная графика

А-С: Напишите программу, которая строит следующий рисунок:

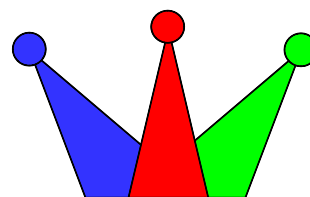
А:



В:



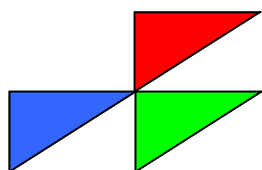
С:



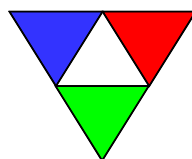
§ 4. Процедуры

А-С: Напишите программу, которая строит следующий рисунок, используя единственную процедуру:

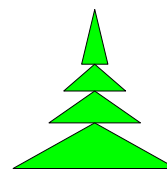
А:



В:



С:



§ 5. Обработка целых чисел

А: Напишите программу, которая вводит трёхзначное число и разбивает его на цифры.

В: Напишите программу, которая вводит с клавиатуры количество секунд и выводит то же самое время в часах, минутах и секундах.

С: Напишите программу, которая вводит с клавиатуры четырёхзначное натуральное число и переставляет его первую и последнюю цифры, например, из числа 1234 должно получиться число 4231.

§ 6. Обработка вещественных чисел

А: Напишите программу, которая вычисляет квадратный корень введённого числа.

В: Напишите программу, которая вводит с клавиатуры координаты двух точек на плоскости и вычисляет длину соединяющего их отрезка.

С: Напишите программу, которая вводит с клавиатуры координаты двух точек в пространстве и вычисляет длину соединяющего их отрезка.

§ 7. Случайные и псевдослучайные числа

А: В игре «Русское лото» из мешка случайным образом выбираются бочонки, на каждом из которых написано число от 1 до 90. Напишите программу, которая выводит наугад первые 5 выигрышных номеров.

В: Напишите программу, которая моделирует бросание двух игральных кубиков: при запуске выводит случайное число на отрезке [2; 12].

С: Игральный кубик бросается три раза (выпадает три случайных значения). Из этих чисел составляется целое число, программа должна найти его квадрат.

§ 8. Ветвления

А: Напишите программу, которая вводит три целых числа и находит максимальное из них.

В: Напишите программу, которая вводит пять целых чисел и находит максимальное из них.

С: Напишите программу, которая вводит последовательно возраст Антона, Бориса и Виктора и определяет, кто из них старше.

§ 9. Сложные условия

А: Напишите программу, которая получает три числа и выводит количество одинаковых чисел в этой цепочке.

В: Напишите программу, которая получает номер месяца и выводит соответствующее ему время года или сообщение об ошибке.

С: Напишите программу, которая получает возраст человека (целое число, не превышающее 120) и выводит этот возраст со словом «год», «года» или «лет». Например, «21 год», «22 года», «25 лет».

§ 10. Циклы с условием

- А:** Напишите программу, которая получает два целых числа A и B ($0 < A < B$) и выводит квадраты всех натуральных чисел на отрезке $[A, B]$.
- В:** Напишите программу, которая получает два целых числа и находит их произведение, не используя операцию умножения. Числа могут быть отрицательными.
- С:** Напишите программу, которая вводит натуральное число N и вычисляет сумму всех чисел Фибоначчи, меньших N . Предусмотреть защиту от ввода отрицательного числа N .

§ 11. Анимация

- А:** Напишите программу, которая моделирует движение объекта, пролетающего через игровое поле.
- В:** Напишите программу, которая моделирует движение двух объекта, пролетающих через игровое поле в разных направлениях.
- С:** Напишите программу, которая моделирует движение двух объекта, пролетающих через игровое поле в разных направлениях. Объекты должны отталкиваться от стенок поля.

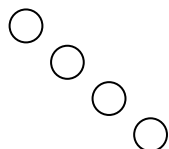
§ 12. Циклы по переменной

- А:** Напишите программу, которая находит все пятизначные числа, которые при делении на 133 дают в остатке 125, а при делении на 134 дают в остатке 111.
- В:** Напишите программу, которая находит все трёхзначные Армстронга (для которых сумма кубов цифр числа равна самому числу).
- С:** Напишите программу, которая получает натуральное число N и выводит на экран все автоморфные числа, не превосходящие N (натуральное число называется автоморфным, если его десятичная запись совпадает с последними цифрами его квадрата, например, $25^2 = 625$).

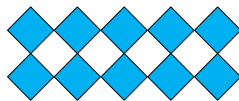
§ 13. Циклы в компьютерной графике

А-С: Напишите программу, которая строит узор:

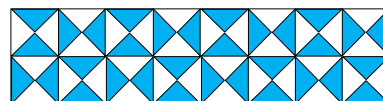
а)



б)



в)



Второй уровень

Программирование на языке Python

§ 1. Проектирование программ

§ 2. Процедуры

- А:** Напишите процедуру, которая принимает параметр – натуральное число N – и выводит на экран квадрат из звёздочек размером $N \times N$ символов.
- В:** Напишите процедуру, которая выводит на экран в столбик все цифры переданного ей числа, начиная с последней.
- С:** Напишите процедуру, которая выводит на экран запись переданного ей числа в римской системе счисления.

§ 3. Рекурсия

- А:** Напишите рекурсивную процедуру, которая переводит число в троичную систему счисления.
- В:** Напишите рекурсивную процедуру, которая переводит число в шестнадцатеричную систему счисления.
- С:** Напишите рекурсивную процедуру, которая переводит число в негавоичную систему счисления (систему с основанием -2).

§ 4. Функции

- А:** Напишите функцию, которая возвращает старшую цифру переданного ей натурального числа.
- В:** Напишите функцию, которая возвращает количество делителей переданного ей натурального числа.
- С:** Напишите логическую функцию, которая возвращает результат *True*, если переданное ей число – это число Фибоначчи.

§ 5. Символьные строки

- А:** Напишите программу, которая заменяет в символьной строке все точки на нули и все буквы «X» на единицы.
- В:** Напишите программу, которая выполняет инверсию битовой строки: заменяет в ней все нули на единицы и наоборот.
- С:** Напишите программу, которая вычисляет сумму неизвестного количества натуральных чисел, записанную в виде символьной строки, например, «1+25+12+34+89»

§ 6. Массивы (списки)

- А:** Напишите программу, которая вводит целое число X и заполняет массив последовательными натуральными числами, начиная с X .
- В:** Напишите программу, которая вводит целое число X и заполняет массив последовательными натуральными числами, начиная с X , в обратном порядке.
- С:** Напишите программу, которая заполняет массив из N элементов первыми N числами Фибоначчи.

§ 7. Алгоритмы обработки массивов

А: Напишите программу, которая определяет сумму элементов массива с чётными значениями.

В: Напишите программу, которая в массиве с чётным количеством элементов меняет местами пары соседних элементов.

С: Напишите программу, которая находит в массиве все числа Фибоначчи и строит из них новый массив.

§ 8. Поиск в массивах

А: Напишите программу, которая находит в массиве минимальный и максимальный элементы и их индексы.

В: Напишите программу, которая находит в массиве количество элементов, равных минимальному, и их индексы.

С: Напишите программу, которая находит в массиве два максимальных элемента и их индексы.

§ 9. Используем массивы

А: Закончите программу из параграфа.

В: Доработайте программу из параграфа так, чтобы тарелки двигались в другом направлении.

С: Доработайте программу из параграфа так, чтобы попадание в меньшую по размеру тарелку приводило к большему увеличению счёта.

§ 10. Матрицы

А: Напишите программу, которая находит максимальный элемент на главной диагонали квадратной матрицы.

В: Напишите программу, которая находит максимальный элемент матрицы и его индексы (номера строки и столбца).

С: Напишите программу, которая выполняет транспонирование матрицы.

§ 11. Сложность алгоритмов

Контроль не предусмотрен.

Третий уровень

Программирование на языке Python

§ 1. Простые алгоритмы сортировки

А: Напишите программу, которая выполняет сортировку массива методом «пузырька».

В: Напишите программу, которая выполняет сортировку массива методом «пузырька» в обратном порядке (сверху вниз).

С: Напишите программу, которая выполняет сортировку двух половин массива (отдельно) методом выбора.

§ 2. Быстрые алгоритмы сортировки

- А:** Напишите программу, которая выполняет сортировку массива по убыванию методом слияния.
- В:** Напишите программу, которая выполняет быструю сортировку массива символьных строк по убыванию длины.
- С:** Напишите программу, которая сравнивает количество операций при сортировке массива методами «пузырька» и слияния.

§ 3. Двоичный поиск

- А:** Напишите программу, которая находит в отсортированном массиве индексы всех элементов, равных заданному значению X .
- В:** Напишите программу, которая сравнивает количество проверок при линейном и двоичном поиске в отсортированном массиве.
- С:** Напишите программу, которая определяет среднее количество проверок при двоичном поиске в отсортированном массиве. Используя результаты работы этой программы, постройте зависимости количества проверок от размера массива.

§ 4. Обработка файлов

- А:** Напишите программу, которая вычисляет сумму чисел, записанных в файле в столбик.
- В:** Напишите программу, которая сортирует числа, записанные в файле в столбик.
- С:** Напишите программу, которая сортирует набор чисел, записанных в двух файлах в столбик. Отсортированные числа должны быть записаны в новый файл.

§ 5. Целочисленные алгоритмы

- А:** Напишите программу, которая вводит натуральное число N и находит все простые числа на отрезке $[2; N]$. Используйте алгоритм «решето Эратосфена».
- В:** Напишите программу, которая вводит натуральные числа M и N и находит все простые числа на отрезке $[M; N]$. Используйте алгоритм «решето Эратосфена».
- С:** Напишите программу, которая определяет количество операций, которые выполняются при решении задачи А. Используя результаты работы этой программы, постройте зависимость количества операций (или времени работы программы) от N .

§ 6. Словари

- А:** Напишите программу, которая строит алфавитно-частотный словарь для заданного файла, в котором каждое слово записано в отдельной строке.
- В:** Напишите программу, которая строит алфавитно-частотный словарь для заданного файла с произвольным текстом.

С: Напишите программу, которая сравнивает тексты разных авторов на основе частоты использования предлогов и союзов.

§ 7. Структуры

А: Напишите программу управления базой данных. Она должна поддерживать добавление, удаление и просмотр записей.

В: Напишите программу управления базой данных. Она должна поддерживать добавление, удаление, просмотр и поиск записей по ключу.

С: Напишите программу управления базой данных. Она должна поддерживать добавление, удаление, просмотр, поиск записей по ключу и сортировку по ключу.

§ 8. Стек, очередь, дек

А: Напишите программу, которая проверяет правильность скобочного выражения с тремя типами скобок.

В: Напишите программу, которая вычисляет значение арифметического выражения, записанного в префиксной форме.

С: Напишите программу, которая применяет алгоритм заливки области для поиска пути в лабиринте.

§ 9. Деревья

А: Напишите программу, которая вычисляет введённое арифметическое выражение без скобок.

В: Напишите программу, которая вычисляет введённое арифметическое выражение со скобками.

С: Напишите программу, которая вычисляет введённое арифметическое выражение со скобками и функциями (*sin*, *cos*, *sqrt*, *abs*).

§ 10. Графы

А: Напишите программу, которая строит минимальное остовное дерево для неориентированного графа.

В: Напишите программу, которая определяет кратчайший маршрут из одной вершины графа в другую с помощью алгоритма Дейкстры.

С: Напишите программу, которая определяет кратчайшие маршрут между каждой парой вершин графа с помощью алгоритма Флойда-Уоршелла.

§ 11. Динамическое программирование

А: Напишите программу, которая вычисляет количество битовых цепочек длины N , в которых нет двух единиц подряд.

В: Напишите программу, которая вычисляет количество битовых цепочек длины N , в которых нет трёх единиц подряд.

С: Напишите программу, которая решает задачу о ранце.

§ 12. Игровые модели

- А:** Напишите программу, которая играет с человеком в игру Баше.
- В:** Напишите программу, которая определяет тип позиции (выигрышная или проигрышная) в игре «крестики-нолики» на доске 3×3.
- С:** Напишите программу, которая играет с человеком в «крестики-нолики».

Четвёртый уровень

Программирование на языке Python

§ 1. Что такое ООП?

§ 2. Модель задачи: классы и объекты

- А:** Перечислите свойства и методы, которыми могли бы обладать объекты класса *Яхта*.
- В:** Постройте объектно-ориентированную модель для задачи моделирования дорожного движения. Рассмотрите классы объектов *Дорога*, *Светофор*, *Машина* и их взаимосвязь.
- С:** Постройте объектно-ориентированную модель для придуманной вами задачи моделирования.

§ 3. Классы и объекты в программе

- А:** Закончите программу, рассмотренную в параграфе.
- В:** Добавьте в программу из параграфа класс *Торпедный аппарат*.
- С:** Добавьте в программу из параграфа классы *Торпедный аппарат* и *Торпеда*.

§ 4. Скрытие внутреннего устройства

- А:** Постройте программную реализацию класса *Перо*. Все данные должны быть скрыты.
- В:** Постройте программную реализацию класса *Машина*. Все данные должны быть скрыты.
- С:** Измените программу игры «Торпедная атака» так, чтобы все поля у объектов были закрытыми.

§ 5. Иерархия классов

- А:** В программе нужно моделировать следующие классы: *Озеро*, *Пресное озеро*, *Солёное озеро*, *Река*, *Море*, *Водоём*, *Океан*. Постройте иерархию классов для этой задачи.
- В:** В программе нужно моделировать следующие классы: *Дерево*, *Осина*, *Берёза*, *Озеро*, *Река*, *Заяц*, *Волк*, *Медведь*. Постройте иерархию классов для этой задачи.
- С:** В программе нужно моделировать следующие классы: *Корабль*, *Подводная лодка*, *Самолет*, *Вертолет*, *Гидросамолет*, *Мотоцикл*, *Трактор*. Постройте иерархию классов для этой задачи.

§ 6. Классы-наследники (I)

A: Закончите программу из параграфа.

B: Добавьте в программу из задания *A* ещё один класс неподвижных объектов, которые изображаются квадратом или ромбом.

C: Добавьте в программу из задания *A* ещё один класс неподвижных объектов, которые при анимации изменяются случайным образом.

§ 7. Классы-наследники (II)

A: Закончите программу из текста параграфа.

B: Измените программу так, чтобы космический корабль, который попал в чёрную дыру или в пульсар, уничтожался. Вместо него должен появляться новый корабль в случайном месте.

C: Добавьте в игру объекты класса *TDestroyer* – специальные боевые космические корабли, которые охотятся за кораблями и странниками и уничтожают их при встрече.

§ 8. Событийно-ориентированное программирование

A: Постройте программу, которая запрашивает разрешение на завершение работы.

B: Измените цвет формы в программе из задания *A*.

C: Доработайте программу из задания *A* так, чтобы при щелчке мышью по форме изменялся цвет формы.

§ 9. Использование компонентов (виджетов)

A: Закончите программу для работы с RGB-кодами цвета.

B: Напишите программу для перевода чисел из восьмеричной системы счисления в десятичную.

C: Напишите программу для перевода чисел из десятичной системы счисления в римскую и обратно.

§ 10. Создание компонентов

A: Постройте компонент для ввода целых чисел.

B: Постройте компонент для ввода целых чисел в шестнадцатеричной системе счисления.

C: Постройте компонент для ввода чисел в римской системе счисления.

§ 11. Модель и представление

A: Закончите программу из параграфа.

B: Постройте программу «Калькулятор» для выполнения вычислений с целыми числами.

C: Постройте программу «Калькулятор» для выполнения вычислений с целыми числами в римской системе счисления.