

МБОУ «Средняя общеобразовательная школа № 176»
ЗАТО г. Зеленогорска Красноярского края

РАССМОТРЕНА
на заседании методического
объединения учителей предметов
точных наук МБОУ «СОШ №176»
Протокол № 1
от «30» августа 2022г.

СОГЛАСОВАНО:
зам. директора по УВР МБОУ
«СОШ №176»
_____ О.А. Доронина
от « 30» августа 2022г.

УТВЕРЖДАЮ:
Директор МБОУ «СОШ №176»
_____ С.А. Дресвянский
Приказ №_110____
от « 30» августа 2022г.

РАБОЧАЯ ПРОГРАММА
Элективного курса по информатике «Основы программирования»
10-11 классов уровня СОО
на 2022 – 2023 учебный год

Учитель: Столбов Константин Александрович

На основании Основной образовательной программы МБОУ «СОШ №176»:

Количество часов в год: 68 (34 – 10 класс, 34 – 11класс)
Количество часов в неделю: 1

2022г.

Пояснительная записка

Рабочая программа элективного курса «Основы программирования» составлена на основе Федерального государственного образовательного стандарта среднего общего образования, основной образовательной программы «МБОУ «СОШ№176, а так же на основе авторской программы К.Ю. Полякова «Программирование на языках Python и C++».

Данный курс реализуется параллельно с углубленным изучением предмета Информатика и призван расширить у обучающихся знания раздела «Программирование».

Проведение практических работ будет происходить на базе детского технопарка «Кванториум».

База для выполнения практических работ в детском технопарке «Кванториум»:

- ноутбуки,
- беспроводной интернет,
- оборудованные места для занятий.

Цели и задачи программы

Цель программы – изучение языков программирования Python и C++.

Задачи программы:

- Развитие интереса обучающихся к изучению программирования;
- Формирование интереса к изучению профессии, связанной с программированием;
- Формирование алгоритмической культуры;
- Формирование навыков грамотной разработки программ, структурного программирования;
- Формирование навыков творческого подхода к решению практических задач спомощью средств программирования.

Планируемые результаты освоения элективного курса

Основные *личностные результаты*, формируемые в процессе освоения программы – это:

- формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершённых творческих учебных проектов;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстративной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки, благодаря реализованным проектам;
- формирование коммуникативной компетенности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- развитие эстетического сознания через творческую деятельность на базе иллюстрированной среды программирования.

К основным **метапредметным** результатам (осваиваемым обучающимися межпредметным понятиям и универсальным учебным действиям, способности их использования как в учебной, так и в познавательной и социальной практике), формируемые в процессе освоения программы курса, можно отнести:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата, понимая, что в программировании длинная программа не значит лучшая программа;
- умение оценивать правильность решения учебно-исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, принятия решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- ИКТ-компетенцию;
- умение сотрудничества и совместной деятельности со сверстниками в процессе проектной и учебно-исследовательской деятельности.

Основные предметные результаты:

В результате изучения курса в 10 классе обучающийся:

- 1) научится применять различные алгоритмы сортировки массивов;
- 2) научится использовать двоичный поиск;
- 3) научится обрабатывать данные, записанные в текстовые и двоичные файлы, сохранять в файлах результаты работы программы;
- 4) научится использовать структуры для объединения данных;
- 5) научится применять словари, стеки, очереди, деки для решения задач обработки данных;
- 6) научится использовать деревья для организации данных;
- 7) познакомится с методами описания графов и некоторыми популярными алгоритмами на графах;
- 8) научится использовать динамическое программирование для решения комбинаторных и оптимизационных задач;
- 9) познакомится с понятием выигрышных и проигрышных позиций в играх с полной информацией.

В результате изучения курса в 11 классе обучающийся:

- 10) познакомится с объектно-ориентированным подходом к разработке программ;
- 11) научится выполнять объектно-ориентированный анализ задачи, выделять свойства и методы объектов;
- 12) научится использовать инкапсуляцию для защиты данных объектов;
- 13) познакомится с понятиями «класс» и «абстрактный класс»;
- 14) познакомится с понятиями «инкапсуляция», «наследование», «полиморфизм»;
- 15) научится проектировать несложные иерархии классов для прикладных задач;
- 16) познакомится с принципами разработки событийно-ориентированных программ;

- 17) научится создавать программы с графическим интерфейсом на языках Python и C#;
- 18) научится использовать готовые и создавать новые компоненты (виджеты) для средбыстрой разработки программ.

Формы организации и режим занятий

Основными формами организации занятий являются традиционные занятия с фронтальной формой работы, практические занятия по решению задач, групповая работа над проектами.

Программа рассчитана на 68 часов (34 часов в 10 классе и 34 часа в 11 классе) из расчета 1 час в неделю.

Результатом обучения считается способность обучающегося 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.

Уровень определяется по результатам выполнения обучающимися практических работ.

Содержание программы

10 класс

Программирование на языке Python

Сортировка массивов. Метод пузырька (сортировка обмeнами). 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.

Программирование на языке C++

Сортировка массивов. Метод пузырька (сортировка простыми обмeнами). Сортировка вставками. Массивы в подпрограммах. Сортировка слиянием. Быстрая сортировка. Стандартная сортировка в языке C++. Двоичный поиск.

Обработка файлов. Файловые потоки. Обработка данных из файла. Чтение текстовых файлов по словам. Построчная обработка файлов. Аргументы основной программы.
Целочисленные алгоритмы. Решето Эратосфена. «Длинные» числа. Динамические массивы. Тип `vector` из библиотеки STL. Итераторы. Словари. Перебор элементов словаря.
Структуры в C++. Обращение к полям структуры. Хранение структур в файлах. Сортировка структур.
Стек. Очередь. Хранение очереди в массиве. Дек.
Деревья в C++. Обходы дерева. Деревья поиска. Вычисление арифметических выражений. Хранение дерева в массиве.
Графы в языке C++. Задача коммивояжера. Жадные алгоритмы. Случайные перестановки. Передача данных по ссылке.
Динамическое программирование. Одномерные задачи. Редактирование строк. Оптимальная стратегия.

11 класс

Программирование на языке Python.

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.

Программирование на языке C++.

Классы и объекты в языке C++. Объектно-ориентированный анализ задачи. Конструкторы классов. Разбиение на модули.
Инкапсуляция. Возможность изменения внутреннего устройства объектов. Свойства «только для чтения».
Наследование. Иерархия классов. Базовый класс. Абстрактный класс. «Чистые» виртуальные методы. Защищённые поля и методы (`protected`).
Полиморфизм. Указатели на базовый класс. Виртуальные методы. Позднее связывание. Деструктор.
Организация взаимодействия объектов.
«Умные» указатели.
RAD-среды для разработки программ. Язык C# и среда .NET. Проект в C#. Свойства объектов. Обработчики событий.
Использование компонентов. Ввод и вывод данных. Обработка ошибок с помощью исключений.
Создание новых классов. Статические методы класса. Создание новых компонентов.

Учебно-тематическое планирование 10 класс

| Номер | Тема занятия | Кол-во часов | | дата | Примечание |
|-------|--|--------------|----------|------|------------|
| | | теория | практика | | |
| | Программирование на языке Python | | | | |
| 1. | Простые алгоритмы сортировки | 0,5 | 0,5 | | |
| 2. | Сортировка слиянием | 0,5 | 0,5 | | |
| 3. | Быстрая сортировка | 0,5 | 0,5 | | |
| 4. | Двоичный поиск | 0,5 | 0,5 | | |
| 5. | Обработка файлов | 0,5 | 0,5 | | |
| 6. | Обработка файлов: практикум | | 1 | | |
| 7. | Целочисленные алгоритмы | 0,5 | 0,5 | | |
| 8. | Словари | 0,5 | 0,5 | | |
| 9. | Структуры | 0,5 | 0,5 | | |
| 10. | Структуры: практикум | | 1 | | |
| 11. | Стек, очередь, дек | 0,5 | 0,5 | | |
| 12. | Деревья | 0,5 | 0,5 | | |
| 13. | Графы | 0,5 | 0,5 | | |
| 14. | Графы: практикум | | 1 | | |
| 15. | Динамическое программирование. | 0,5 | 0,5 | | |
| 16. | Динамическое программирование: практикум | | 1 | | |
| 17. | Игровые модели | 0,5 | 0,5 | | |
| 18. | Игровые модели: практикум | | 1 | | |
| | Программирование на языке C++ | | | | |
| 19. | Простые алгоритмы сортировки | 0,5 | 0,5 | | |

| | | | | | |
|-----|---|------|------|--|--|
| 20. | Быстрые алгоритмы сортировки и поиска | 0,5 | 0,5 | | |
| 21. | Обработка файлов | 0,5 | 0,5 | | |
| 22. | Целочисленные алгоритмы | 0,5 | 0,5 | | |
| 23. | Динамические массивы и словари | 0,5 | 0,5 | | |
| 24. | Итераторы | 0,5 | 0,5 | | |
| 25. | Структуры | 0,5 | 0,5 | | |
| 26. | Структуры: практикум | | 1 | | |
| 27. | Стек, очередь, дек | 0,5 | 0,5 | | |
| 28. | Деревья | 0,5 | 0,5 | | |
| 29. | Графы | 0,5 | 0,5 | | |
| 30. | Графы: практикум | | 1 | | |
| 31. | Динамическое программирование | 0,5 | 0,5 | | |
| 32. | Динамическое программирование:практикум | | 1 | | |
| 33. | Работа над проектами. Практикум. | | 1 | | |
| 34. | Работа над проектами. Защита.Анализ. | | 1 | | |
| | | 11,5 | 22,5 | | |

Учебно-тематическое планирование 11 класс

| номер | Тема занятия | Кол-во часов | | Дата | Примечание |
|-------|--|--------------|----------|------|------------|
| | | теория | практика | | |
| | Программирование на языке Python | | | | |
| 1. | Что такое ООП? | 1 | | | |
| 2. | Модель задачи: классы и объекты | 0,5 | 0,5 | | |
| 3. | Классы и объекты в программе | 0,5 | 0,5 | | |
| 4. | Классы и объекты в программе: практикум | | 1 | | |
| 5. | Скрытие внутреннего устройства | 0,5 | 0,5 | | |
| 6. | Иерархия классов | 0,5 | 0,5 | | |
| 7. | Классы-наследники (I) | 0,5 | 0,5 | | |
| 8. | Классы-наследники (II) | 0,5 | 0,5 | | |
| 9. | Доработка игры | | 1 | | |
| 10. | Событийно-ориентированное программирование | 0,5 | 0,5 | | |
| 11. | Использование компонентов (виджетов) | 1 | | | |
| 12. | Использование компонентов (виджетов): практикум. | | 1 | | |
| 13. | Создание компонентов | 0,5 | 0,5 | | |
| 14. | Модель и представление | 0,5 | 0,5 | | |
| 15. | Выполнение проекта. Практикум. | | 1 | | |
| 16. | Выполнение проекта. Защита. Анализ. | | 1 | | |
| | Программирование на языке C++ | | | | |
| 17. | Классы и объекты | 0,5 | 0,5 | | |
| 18. | Программа с классами | 0,5 | 0,5 | | |

| | | | | | |
|-----|---|------|------|--|--|
| 19. | Программа с классами: практикум | | 1 | | |
| 20. | Инкапсуляция | 0,5 | 0,5 | | |
| 21. | Наследование | 0,5 | 0,5 | | |
| 22. | Наследование: практикум | | 1 | | |
| 23. | Полиморфизм | 0,5 | 0,5 | | |
| 24. | Полиморфизм: практикум | | 1 | | |
| 25. | Взаимодействие объектов | 0,5 | 0,5 | | |
| 26. | Простая программа на C# | 0,5 | 0,5 | | |
| 27. | Использование компонентов | 0,5 | 0,5 | | |
| 28. | Ввод и вывод данных | 0,5 | 0,5 | | |
| 29. | Создание новых классов | 0,5 | 0,5 | | |
| 30. | Выполнение проекта. Проектирование. | | 1 | | |
| 31. | Выполнение проекта. Формализация. | | 1 | | |
| 32. | Выполнение проекта. Практикум. | | 1 | | |
| 33. | Выполнение проекта. Компьютерный эксперимент. | | 1 | | |
| 34. | Выполнение проекта. Защита. Анализ. | | 1 | | |
| | | 11,5 | 22,5 | | |

Учебно-методическое обеспечение

1. Андреева Е.В. Программирование – это так просто, программирование – это так сложно, МЦНМО, 2015
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. Вильямс: М., 2017
3. Лутц М. Изучаем Питон (4-е издание). Символ-плюс: М., 2017
4. Поляков К.Ю. Программирование. Python и C++. Учебное пособие для общеобразовательных организаций. В 4 частях. Ч. 1-4 – М.: БИНОМ. Лаборатория знаний, 2019
5. Прата С. Язык программирования C++. Лекции и упражнения. – Диалектика. Вильямс, 2016
6. Программа курса «Программирование. Python и C++» 8-11 класс (профильная школа) <http://kpolyakov.spb.ru/school/pycpp.htm>.

Информационно-образовательные ресурсы

<http://kpolyakov.spb.ru/school/pycpp.htm> - страница автора учебного пособия К.Ю. Полякова «Программирование. Python и C++» - тексты программ, дополнительные материалы, ссылки на необходимое программное обеспечение;

<https://informatics.mccme.ru/course/view.php?id=666> – автоматическая проверка и тестирование программ, платформа для проведения конкурсов.

<https://silvertests.ru/> - автоматическая проверка и тестирование программ.