

ДЕПАРТАМЕНТ ОБРАЗОВАНИЯ И НАУКИ ГОРОДА МОСКВЫ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ГОРОДА МОСКВЫ
«ШКОЛА № 1501»

127055, г. Москва, Тихвинский пер., д. 3
ИНН/КПП 7707290995/770701001, ОГРН 1027700509803, ОКПО 53793153

тел./факс (499)973-02-84, e-mail: 1501@edu.mos.ru
Сайт: http://lycc1501.mskobr.ru/

ПРИНЯТО
методическим объединением
учителей информатики
ГБОУ Школа № 1501

Протокол № 4
от «23» мая 20 19 г.

«УТВЕРЖДАЮ»
Директор
ГБОУ Школа № 1501

Приказ № 182/3 019
от «23» мая 20 19 г.

/ Н.Т. Рахимова /



Рабочая программа по предмету Информатика (специальный курс)

Образовательный проект
«ИТ-класс в московской школе»
(направление «Программирование»)

Москва, 2019г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

ВВЕДЕНИЕ

Настоящая рабочая программа разработана в рамках образовательного проекта «ИТ-класс в московской школе» коллективом авторов ФГАОУ ВО «Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики» под руководством Хлопотова М.В., (к.т.н., доцент ФИКТ). Соавторы: Зубок Д.А., (к.ф.-м.н., доцент ФИТиП), Маятин А.В. (к.пед.н., доцент ФИТиП).

Курс «ПРОГРАММИРОВАНИЕ» является частью образовательной программы для ИТ-классов средней школы.

Элементы обучения (языки программирования, программное обеспечение средств вычислительной техники и автоматизированных систем, математические и алгоритмические модели программных систем и комплексов, методы и инструменты разработки и тестирования программного продукта, процессы жизненного цикла программного продукта) вводятся с первого полугодия 10 класса с постепенным усложнением содержания соответственно возрасту обучающегося и заканчиваются во втором полугодии 11-го класса.

Курс носит междисциплинарный характер и может быть фактически разнесен между часами, отведенными на элективные дисциплины и внеурочную деятельность.

Предлагаемая программа соответствует положениям федерального государственного образовательного стандарта среднего общего образования.

Программа курса отражает способы формирования универсальных учебных действий, составляющих основу для профессионального самоопределения, саморазвития и непрерывного образования, выработки коммуникативных качеств, целостности общекультурного, личностного и познавательного развития учащихся.

Рабочая программа составлена на основе:

- Закона об образовании Российской Федерации
- Федерального государственного образовательного стандарта среднего общего образования.
- Профессиональных стандартов: 06.001 Программист, 06.028 Системный программист, 06.035 Разработчик Web и мультимедийных приложений

Программа соответствует требованиям к структуре программ, заявленным в ФГОС, и включает следующие разделы:

- Пояснительная записка, в которой уточняются общие цели образования с учетом специфики курса.
- Общая характеристика курса, содержащая ценностные ориентиры образования по профилю «Программирование».
- Место данного курса в учебном плане.
- Результаты освоения курса (личностные, метапредметные и предметные), соответствующие глобальным целям образования по профилю «Программирование» и принципу развивающего обучения, лежащему в основе предлагаемой программы.
- Содержание курса по направлению «Программирование» в 10 и 11 классах.
- Тематическое планирование, которое дает представление об основных видах учебной деятельности в процессе освоения курса в 10-11 классах основной школы.
- Рекомендации по учебно-методическому и материально-техническому обеспечению образовательного процесса.
- Планируемые результаты освоения программы.

ЦЕЛИ И ЗАДАЧИ РЕАЛИЗАЦИИ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ ОСНОВНОГО ОБЩЕГО ОБРАЗОВАНИЯ ПО КУРСУ

Целями курса является формирование у обучающегося алгоритмического мышления и, соответственно, необходимых знаний и умений, необходимых для успешного развития в направлении дальнейшей деятельности в области программирования.

Для достижения поставленных целей образование в области разработки программных средств призвано обеспечить решение следующих задач:

- формирование в процессе решения практических задач у учащихся навыков алгоритмического мышления и понимания средств формального описания алгоритмов;
- овладение приёмами написания программ на языках программирования с использованием основных конструкций;
- осознание практической применимости выполняемых учебных задач в современном обществе для возможного выбора этой области в качестве будущей профессии.

ПРИНЦИПЫ И ПОДХОДЫ К ФОРМИРОВАНИЮ ПРОГРАММЫ

Реализация принципа системности, обеспечение связности профиля «программирование» с другими разделами и темами информатики.

Самым важным принципом в процессе обучения программированию является решение практических задач и участие в учебных проектах. В результате освоения успешного программы ученики смогут понять принципы большинства современных языков программирования и применить на практике полученные знания.

СОСТАВ УЧАСТНИКОВ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

Программа основного общего образования рассчитана на реализацию в 10 - 11 классах общеобразовательных учреждений и учреждений с углубленным изучением отдельных предметов, и нацелена на возрастную категорию учащихся 15 – 18 лет.

ОБЩАЯ ХАРАКТЕРИСТИКА УЧЕБНОГО КУРСА

Представленная программа направления «Программирование» (10-11 класс)» предназначена для практического освоения учащимися следующих видов профессиональной деятельности:

- Разработка и отладка программного кода
- Тестирование и рефакторинг программного кода
- Разработка требований и проектирование программного обеспечения
- Интеграция программных модулей и компонент и верификация выпусков программного продукта

Программа рассчитана на 2 года (10-11 класс), при этом обучение можно условно разделить на 4 модуля:

- Модуль 1 посвящён практике программирования;
- Модуль 2 посвящён алгоритмам и структурам данных;
- Модуль 3 посвящён основам проектирования программного обеспечения;
- Модуль 4 посвящён разработке приложений, интегрированных в ИТ-инфраструктуру.

Содержание курса направлено на формирование универсальных учебных действий, обеспечивающих развитие познавательных и коммуникативных качеств личности. Обучающиеся включаются в деятельность, основу которой составляют такие учебные действия, как умение видеть проблемы, ставить вопросы, классифицировать, наблюдать, делать выводы, объяснять, доказывать, защищать свои идеи, давать определение понятиям, структурировать материал и др. обучающиеся включаются в коммуникативную учебную деятельность, где

преобладают такие ее виды, как умение полно и точно выразить свои мысли, аргументировать свою точку зрения, работать в группе, представлять и сообщать информацию в устной и письменной форме, вступать в диалог и др.

Программа курса «Программирование» для средней школы предусматривает реализацию следующих принципов:

Придать развитию знаний динамичный характер: использовать ранее полученные знания при овладении новыми понятиями, постепенно углублять и развивать ведущие понятия в процессе изучения всего курса.

Сконцентрировать учебный материал, укрупнив комплектные единицы знаний, что создает дидактические условия для развития системного мышления у учащихся: освободить учебный материал от деталей, имеющих специальное значение, но излишних для общего образования, группируя при этом частные понятия, необходимые для общего образования, вокруг ведущих понятий.

Формировать у обучающихся системное мышление, сочетая его с активной познавательной деятельностью обучающихся.

Учитывать возрастные, индивидуальные особенности и возможности обучающихся, предлагая им задания по выбору.

ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ УЧЕБНОГО КУРСА

Деятельность образовательного учреждения в обучении по направлению «Программирование» должна быть направлена на достижение обучающимися следующих **личностных результатов**:

- формирование алгоритмического мышления;
- формирование интеллектуальных умений: анализировать информацию, анализировать основные изученные понятия, строить рассуждения, анализировать и сопоставлять теоретические знания с их практической применимостью;
- готовность и способность к самостоятельной, творческой и ответственной деятельности;
- навыки сотрудничества со сверстниками, детьми младшего возраста, взрослыми в образовательной, общественно полезной, учебно-исследовательской, проектной и других видах деятельности;
- готовность и способность к образованию, в том числе самообразованию;
- сознательное отношение к непрерывному образованию как условию успешной профессиональной и общественной деятельности;
- осознанный выбор будущей профессии и возможностей реализации собственных жизненных планов;
- отношение к профессиональной деятельности как возможности участия в решении личных, общественных, государственных, общенациональных проблем;
- понимание роли информационных процессов в современном мире;
- владение первичными навыками анализа и критичной оценки получаемой информации;

Метапредметными результатами освоения программы по направлению «Программирование» являются:

- знание общепредметных понятий: информация, данные, алгоритм, исполнитель, программа, программирование, языки программирования, система, функция, объект,
- владение универсальными умениями: постановка задачи, формулирование проблемы; поиск, выделение и структурирование необходимой информации; выбор наиболее эффективных методов решения задачи в зависимости от конкретных условий; самостоятельное создание алгоритмов решения задач;
- умение применить изученные понятия для реализации учебных задач;

- умение анализировать имеющийся инструментарий и применять его к поставленной задаче;
- умение анализировать результат своей предыдущей деятельности и приводить его к виду, требуемому на следующем этапе;
- умение работать с различными источниками информации, применять на практике полученные знания, анализировать модели;

Предметные результаты освоения программы по информатике приведены в разделе 3 (Поурочно-тематическое планирование) рабочей программы для каждого отдельного урока.

ЗАКЛЮЧЕНИЕ

Таким образом, данная учебная программа удовлетворяет требованиям Федерального государственного образовательного стандарта среднего общего образования по изучению дополнительных учебных предметов, курсов по выбору обучающихся.

2. СОДЕРЖАНИЕ КУРСА

Рабочей программой предусмотрен следующий тематический план, который представлен в таблице 1.

Таблица 1 Тематический план

№ п/п	Модуль	Наименование раздела	Количество часов
1 полугодие 10 класса			
1	Алгоритмы и структуры данных	Оценка сложности алгоритмов на примере алгоритмов сортировки	8
2	Алгоритмы и структуры данных	Элементарные структуры данных	8
3	Практика программирования	Работа со строками, файлами и графикой	8
4	Алгоритмы и структуры данных	Алгоритмы поиска	8
2 полугодие 10 класса			
5	Алгоритмы и структуры данных	Деревья поиска	8
6	Алгоритмы и структуры данных	Хеширование	7
7	Практика программирования	Решение олимпиадных задач по программированию и алгоритмизации	9
8	Практика программирования	Совместная работа над проектом с использованием системы контроля версий	8
1 полугодие 11 класса			
9	Алгоритмы и структуры данных	Целочисленные алгоритмы. Использование связанных структур. Графы. «Жадные» алгоритмы. Алгоритм Дейкстры. Динамическое программирование.	8
10	Практика программирования	Проверка и отладка программного кода Разработка процедур проверки работоспособности и измерения характеристик программного обеспечения Проверка работоспособности программного обеспечения	8
11	Основы проектирования программного обеспечения	Жизненный цикл программного обеспечения Качество программного обеспечения Анализ требований к программному обеспечению Документирование программного обеспечения	8
12	Разработка приложений, интегрированных в ИТ-инфраструктуру	Подключение и взаимодействие с системами управления базами данных Протоколы передачи данных между компонентами клиент-серверной архитектуры Разработка клиент-серверных приложений	8
2 полугодие 11 класса			
13	Практика программирования	Основы объектно-ориентированного программирования. Рефакторинг и оптимизация программного кода	8

14	Основы проектирования программного обеспечения	Разработка технических спецификаций на программные компоненты и их взаимодействие Обеспечение качества программных систем Основные принципы проектирования программного обеспечения	8
15	Разработка приложений, интегрированных в ИТ-инфраструктуру	Особенности работы приложений под управлением различных операционных систем, сборка дистрибутивов приложений Защищенное хранение данных в файлах с ограниченным доступом в различных операционных системах Подготовка и защита этапов учебного проекта	16

Примечание. Разделы, относящиеся к одному модулю, могут быть реализованы в различных полугодиях. В том числе, возможно параллельное изучение материала нескольких модулей, если это обосновано логикой освоения материала.

Краткое содержание разделов:

- 1. Оценка сложности алгоритмов на примере алгоритмов сортировки.** *Оценка сложности алгоритмов.* Необходимость оценки сложности программ. Порядок роста. Константная, линейная, логарифмическая, квадратичная сложность. Наилучший, средний и наихудший случай. Оценка времени работы алгоритмов. Оценка алгоритма по памяти. Анализ алгоритмов с ветвлениями и циклами. *Алгоритмы сортировки.* Алгоритмы сортировки, основанные на сравнении: сортировка слиянием, быстрая сортировка. Оценка сложности алгоритмов сортировки. Оценка время работы алгоритмов сортировки.
- 2. Элементарные структуры данных.** Стек. Использование списка. Вычисление арифметических выражений с помощью стека. Проверка скобочных выражений. Очереди, деки.
- 3. Работа со строками, файлами и графикой.** *Символьные строки.* Операции со строками. Поиск в строках. Примеры обработки строк. Преобразование число-строка. Строки в процедурах и функциях. Рекурсивный перебор. *Работа с файлами.* Работа с текстовым файлом: чтение, запись, дозапись. Работа с бинарными файлами. *Работа с графикой.* Библиотеки для работы с графикой. Графические примитивы. Основные функции работы с графикой. Решение задач на моделирование.
- 4. Алгоритмы поиска.** Последовательный поиск. Двоичный поиск в отсортированном массиве. Оценка сложности алгоритмов поиска.
- 5. Деревья поиска.** Деревья. Двоичные деревья поиска. Обход дерева поиска. Поиск элемента. Поиск минимума и максимума. Вставка и удаление элементов. *Сбалансированные деревья поиска.* Обзор сбалансированных деревьев, AVL-дерево, Sprau-дерево. Операции со сбалансированными деревьями поиска.
- 6. Хеширование.** Хеш-таблицы с закрытой и открытой адресацией. Свойства хеш-таблицы. Хеширование в современных языках программирования.
- 7. Решение олимпиадных задач.** Решение и разбор задач. Типичные ошибки в программе и их исправление.
- 8. Системы контроля версий. Совместная работа над проектом.** *Рецензирование кода.* Оформление программного кода в соответствии с установленными требованиями. Руководство по стилю. Техники рецензирования кода. Ветки в Git. Совместная работа. Проект на Github. Fork и Pull Request.

3. ПОУРОЧНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

№ п/п	Номер урока	Тема урока	Изучаемые элементы содержания	Планируемые предметные результаты обучения (контролируемые элементы содержания)
1	1-3	Оценка сложности алгоритмов	Изучение понятия сложность алгоритма. Необходимость оценки сложности программ. Изучение подхода к оценке сложности алгоритмов. Изучение понятия «порядок роста», константная, линейная, логарифмическая, квадратичная сложность. Примеры оценки сложности алгоритмов. Наилучший, средний и наихудший случай при оценке сложности алгоритма. Оценка времени работы алгоритмов. Оценка алгоритма по памяти. Анализ алгоритмов с ветвлениями и циклами.	3.4.1 Формализация понятия алгоритма. Машина Тьюринга 3.4.5 Сложность вычисления 3.4.6 Задачи анализа алгоритмов: определение входных данных, при которых алгоритм дает указанный результат 3.4.7 Задачи анализа алгоритмов: определение результата алгоритма без его полного пошагового выполнения.
2	4	Оценка сложности алгоритмов	Решение задачи двумя способами: неэффективным и эффективным. Оценка сложности алгоритмов.	3.1.1 Алгоритмические конструкции 3.1.2 Разработка и программная реализация алгоритмов решения типовых задач базового уровня 3.4.5 Сложность вычисления
3	5	Алгоритмы сортировки	Реализация алгоритма сортировки. Оценка сложности реализованного алгоритма.	3.2.6 Табличные величины. Массивы 3.1.4 Рекурсивные алгоритмы. Сортировка массивов. 3.4.5 Сложность вычисления
4	6-7	Алгоритмы сортировки	Разбор алгоритмов сортировки, основанных на сравнении: сортировка слиянием, быстрая сортировка. Оценка сложности алгоритмов сортировки. Оценка времени работы алгоритмов сортировки.	3.1.4 Рекурсивные алгоритмы. Сортировка массивов. 3.4.5 Сложность вычисления
5	8	Оценка сложности алгоритмов на примере алгоритмов сортировки	Тест с заданиями по темам «Оценка сложности алгоритмов», «Алгоритмы сортировки».	3.1.1 Алгоритмические конструкции 3.1.4 Рекурсивные алгоритмы. Сортировка массивов. 3.4.5 Сложность вычисления
6	9	Элементарные структуры данных.	Изучение понятия динамические структуры данных. Изучение понятия стек. Реализация стека с использованием массива и списка.	3.1.8 Динамические структуры данных и операции с ними
7	10-11	Элементарные структуры данных	Реализация операций по работе со стеком: добавление, удаление, просмотр вершины без удаления, очистка стека.	3.1.8 Динамические структуры данных и операции с ними
8	12	Элементарные структуры данных	Вычисление арифметических выражений с помощью стека. Проверка скобочных выражений.	3.1.1 Алгоритмические конструкции 3.1.8 Динамические структуры данных и операции с ними
9	13	Элементарные структуры данных	Изучение понятия динамические структуры данных. Изучение понятия очередь. Реализация очереди с использованием массива и списка. Сравнение очереди и стека.	3.1.8 Динамические структуры данных и операции с ними
10	14	Элементарные структуры данных	Реализация операций по работе с очередью: добавление в хвост, удаление из головы, очистка очереди.	3.1.1 Алгоритмические конструкции 3.1.8 Динамические структуры данных и операции с ними
11	15	Элементарные структуры данных	Изучение структуры данных дек (двусторонняя очередь). Реализация простого дека. Реализация операций по работе с деком.	3.1.8 Динамические структуры данных и операции с ними
12	16	Элементарные структуры данных	Решение задач по теме «Элементарные структуры данных».	3.1.8 Динамические структуры данных и операции с ними
13	17	Символьные строки	Символ и строка. Операции со строками. Написание программ для поиска и подсчёта символов в строках. Примеры обработки строк. Преобразование число-строка.	3.1.5 Алгоритмы анализа символьных строк
14	18	Символьные строки	Написание программ с использованием строк в процедурах и функциях. Реализация рекурсивного перебора.	3.1.5 Алгоритмы анализа символьных строк 3.2.5 Рекурсивные процедуры и функции
15	19	Работа с файлами	Работа с текстовым файлом: чтение, запись, дозапись. Работа с бинарными файлами.	3.2.7 Средства работы с данными во внешней памяти. Файлы.
16	20-21	Работа с файлами	Выполнение заданий по темам «Ветвления», «Циклические алгоритмы», «Процедуры», «Функции», «Рекурсия и рекурсивные алгоритмы», «Массивы»: входные данные	3.1.2 Разработка и программная реализация алгоритмов решения типовых задач базового уровня

			читаются из файла, результат записывается в другой файл.	3.2.7 Средства работы с данными во внешней памяти. Файлы. 3.3.11 Решение задач из различных предметных областей с использованием выбранного языка программирования
17	22-23	Работа с графикой	Изучение возможностей библиотек для работы с графикой. Построение графических примитивов. Изучение основных функций работы с графикой. Решение задач на моделирование.	3.1.1 Алгоритмические конструкции
18	24	Работа со строками, файлами и графикой	Решение задач по темам «Символьные строки», «Работа с файлами», «Работа с графикой»	3.1.2 Разработка и программная реализация алгоритмов решения типовых задач базового уровня 3.4.8 Решение задач из различных предметных областей с использованием выбранного языка программирования
20	25-26	Алгоритмы поиска	Постановка задачи поиска элемента в массиве. Изучение алгоритмов поиска: последовательный поиск. Реализация последовательного поиска.	3.1.3 Алгоритмы обработки массивов.
21	27-28	Алгоритмы поиска	Повторение понятия рекурсии и алгоритмов сортировки. Изучение алгоритма двоичного поиска в отсортированном массиве. Написание программы, реализующий двоичный поиск.	3.1.3 Алгоритмы обработки массивов. 3.1.4 Рекурсивные алгоритмы. Сортировка массивов. 3.2.5 Рекурсивные процедуры и функции
22	29-30	Алгоритмы поиска	Написание программы для сравнения последовательного и двоичного поиска. Оценка сложности алгоритмов поиска.	3.1.3 Алгоритмы обработки массивов. 3.1.4 Рекурсивные алгоритмы. Сортировка массивов. 3.2.5 Рекурсивные процедуры и функции 3.4.5 Сложность вычисления
23	31	Алгоритмы поиска	Решение задач по разделу	3.1.3 Алгоритмы обработки массивов. 3.1.4 Рекурсивные алгоритмы. Сортировка массивов.
24	32	Алгоритмы поиска	Решение задач по разделу	3.1.3 Алгоритмы обработки массивов. 3.1.4 Рекурсивные алгоритмы. Сортировка массивов.
25	33-35	Деревья поиска	Деревья. Двоичные деревья поиска. Обход дерева поиска. Поиск элемента. Поиск минимума и максимума. Вставка и удаление элементов.	2.5.4 Деревья. Основные понятия: корень, вершина (узел), лист, поддерево 2.5.1 Решение алгоритмических задач, связанных с анализом графов
26	36-37	Деревья поиска	Сбалансированные деревья поиска. Обзор сбалансированных деревьев, AVL-дерево, Splay-дерево	2.5.4 Деревья. Основные понятия: корень, вершина (узел), лист, поддерево 2.5.1 Решение алгоритмических задач, связанных с анализом графов
27	38-39	Деревья поиска	Операции со сбалансированными деревьями поиска.	2.5.4 Деревья. Основные понятия: корень, вершина (узел), лист, поддерево 2.5.1 Решение алгоритмических задач, связанных с анализом графов
28	40	Деревья поиска	Решение задач по разделу	2.5.4 Деревья. Основные понятия: корень, вершина (узел), лист, поддерево 2.5.1 Решение алгоритмических задач, связанных с анализом графов
26	41-42	Хеширование	Кодирование информации. Необратимый код. Применение хеш-функций: криптография, ускорение поиска данных. Понятие коллизии, методы борьбы с коллизиями. Виды хеш-функций.	2.1.1 Знак. Знаковые системы. Способы кодирования информации 3.2.2 Типы и структуры данных. Операции с ними. 3.3.5 Вспомогательные алгоритмы и программы. Процедуры (подпрограммы) и библиотеки подпрограмм. 3.1.9 Хеширование
27	43-44	Хеширование	Хеш-таблицы с закрытой и открытой адресацией. Свойства хеш-таблицы.	3.1.9 Хеширование

				3.2.2 Типы и структуры данных. Операции с ними. 3.3.5 Вспомогательные алгоритмы и программы. Процедуры (подпрограммы) и библиотеки подпрограмм.
28	45	Хеширование	Хеширование в современных языках программирования	3.1.9 Хеширование
29	46	Хеширование	Решение задач по разделу	3.1.9 Хеширование
30	47	Хеширование	Решение задач по разделу	3.1.9 Хеширование
31	48-52	Решение олимпиадных задач	Разбор олимпиадных задач. Решение задач с применением изученных средств языка программирования.	3.3.11. Решение задач из различных предметных областей с использованием выбранного языка программирования
32	53-54	Решение олимпиадных задач	Разбор олимпиадных задач. Решение задач с применением изученных алгоритмов и структур данных.	3.3.11. Решение задач из различных предметных областей с использованием выбранного языка программирования
33	55	Решение олимпиадных задач	Решение олимпиадных задач, с применением знаний и умений по изученным разделам.	3.3.11. Решение задач из различных предметных областей с использованием выбранного языка программирования
34	56	Решение олимпиадных задач	Разбор типичных ошибок, допущенных в контрольной работе.	3.3.11. Решение задач из различных предметных областей с использованием выбранного языка программирования
35	57-58	Системы контроля версий. Совместная работа над проектом.	<i>Рецензирование кода.</i> Оформление программного кода в соответствии с установленными требованиями. Руководство по стилю. Техники рецензирования кода.	3.3.1 Этапы решения задач на компьютере. 3.3.11 Решение типовых задач. Проверка работоспособности программы с использованием трассировочных таблиц
36	59	Системы контроля версий. Совместная работа над проектом.	Системы контроля версий. Совместная работа.	3.3.1 Этапы решения задач на компьютере. 3.3.19 Системы контроля версий
37	60-63	Системы контроля версий. Совместная работа над проектом.	Ветки в Git. Создание проекта на Github. Fork и Pull Request.	3.3.1 Этапы решения задач на компьютере. 3.3.19 Системы контроля версий
38	64	Системы контроля версий. Совместная работа над проектом.	Защита проекта	3.3.1 Этапы решения задач на компьютере. 3.3.19 Системы контроля версий

4. УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА.

Оборудование

1. Рабочая станция ученика (Intel i5, 8Гб ОЗУ, SSD 500Гб, видеокарта Radeon RX VEGA 56 или аналогичная с 8Гб видеопамяти, монитор с разрешением 1920x1080, клавиатура, мышь)
2. Рабочая станция учителя (Intel i7, 16Гб ОЗУ, SSD 500Гб, видеокарта Radeon RX VEGA 56 или аналогичная с 8Гб видеопамяти, монитор с разрешением 1920x1080, клавиатура, мышь)
3. Мобильный класс из MacBook Air (2-ядерный процессор с тактовой частотой Intel Core i5 1,6 ГГц, ускорение Turbo Boost до 3,6 ГГц, накопитель 128 Гб SSD, 8Гб ОЗУ)
4. Планшетный класс из iPad
5. Сервер (1U, 14-ядерный Intel Xeon E5-2680V4 2400 МГц, 16 Гб DDR-4, 4 x 2.5" SATA/SAS, 4xGigabit Ethernet (1000 Мбит/с), 750 Вт, ОС Windows Server с предустановленной ActiveDirectory, RAID-1 массив из 2-х HDD 3Тб)
6. Источник бесперебойного питания для сервера (ИБП)
7. Телекоммуникационный шкаф
8. Коммутаторы уровня доступа с поддержкой VLAN и IP-телефонии
9. Копировальная станция
10. Интерактивная панель (например IRBIS “Панель МЭШ”)

Программное обеспечение (в том числе системное ПО)

1. ОС Windows 10
2. MS Office 2016
3. VisualStudio 2017
4. Oracle SQL Developer
5. MySQL
6. PyCharm CE
7. Python 3.7
8. XAMPP
9. Антивирус Касперского

Список учебной и методической литературы и другие источники

1. Информатика. 10 класс. Базовый и углубленный уровни : учебник : в 2 ч. Ч. 1. ФГОС / К. Ю. Поляков, Е. А. Еремин. — М. : БИНОМ. Лаборатория знаний, 2018. — 352 с. : ил.

2. Информатика. 10 класс. Базовый и углубленный уровни : учебник : в 2 ч. Ч. 2. ФГОС / К. Ю. Поляков, Е. А. Еремин. — М. : БИНОМ. Лаборатория знаний, 2018. — 352 с. : ил.
3. Информатика. 11 класс. Базовый и углубленный уровни : учебник: в 2 ч. Ч. 1. ФГОС / К. Ю. Поляков, Е. А. Еремин. — М. : БИНОМ. Лаборатория знаний, 2019. — 232 с. : ил.
4. Информатика. 11 класс. Базовый и углубленный уровни : учебник: в 2 ч. Ч. 2. ФГОС / К. Ю. Поляков, Е. А. Еремин. — М. : БИНОМ. Лаборатория знаний, 2019. — 302 с. : ил.
5. Поляков К. Ю. Программирование. Python. C++. Часть 1. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 144 с.
6. Поляков К. Ю. Программирование. Python. C++. Часть 2. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 176 с.
7. Поляков К. Ю. Программирование. Python. C++. Часть 3. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 208 с.
8. Поляков К. Ю. Программирование. Python. C++. Часть 4. Учебное пособие — М. : БИНОМ. Лаборатория знаний, 2019. — 192 с.
9. Грэхэм Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики: Пер. с англ. – М. Мир, 1998. – 703 с., с илл.
10. Дейт К. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1328 с.: ил.
11. Вирт Н. Алгоритмы и структуры данных. – М.: ДМК Пресс, 2010. – 272 с.
12. Кормен Т., Лейзерсон Ч., Ривест Р, Штайн К. Алгоритмы. Построение и анализ — М., Вильямс, 2011 г. — 1296 стр.
13. Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд.: Пер. с англ. — СПб.: ООО “Альфа-книга”, 2017. — 432 с. : ил.
14. Златопольский Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2018. – 284 с.
15. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»)
16. Зубок Д.А., Маятин А.В., Краснов С.В. Основы программирования в среде TURBO PASCAL. Учебное пособие. – СПб: СПбГУ ИТМО; 2009. - 69 с.
17. Петров В.Ю. Информатика. Алгоритмизация и программирование. Часть 1: Учебное пособие. - Санкт-Петербург: Университет ИТМО, 2016. - 91 с.
18. Лебедев И.С., Петров В.Ю. Информатика. Программирование. Часть 2: Учебно-методическое пособие. - Санкт-Петербург: Университет ИТМО, 2016. - 71 с.
19. Осипов Н.А. Технологии программирования: Учебное пособие. - Санкт-Петербург: Университет ИТМО, 2016. - 61 с.

5. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ИЗУЧЕНИЯ УЧЕБНОГО КУРСА

Выпускник научится:

- 1) преобразовывать абстрактную идею в последовательность конкретных шагов, необходимых для её воплощения на практике;
- 2) выполнять дискретизацию задачи как необходимый шаг для её решения с помощью компьютера;
- 3) составлять и записывать алгоритм с помощью определенных средств и методов описания; оценивать алгоритмы; применять на практике основные алгоритмические структуры – линейную, условную и циклическую; разбивать сложные задачи на подзадачи;
- 4) записывать алгоритмы обработки данных на изучаемом языке программирования (из перечня: Паскаль, Python, C, C++, C#), отлаживать и выполнять полученную программу в используемой среде разработки;
- 5) использовать информационные и коммуникационные технологии для хранения, преобразования и передачи различных видов информации.