

Образовательная программа «КОМПЬЮТЕРНЫЕ НАУКИ»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа курса "Компьютерные науки" составлена на основе опыта, накопленного в результате преподавания аналогичного предмета в старших классах системы школьного образования Израиля на протяжении последних 35 лет и школах Пензенской области в течении 8 лет. Учтены также требования современной сферы рынка высоких технологий (хай-тека) к знаниям выпускников школ, как к потенциальным активным участникам деятельности этой сферы в рамках национальной экономики и международных.

Аннотация

Компьютеризация и информатизация стали сегодня естественными составляющими функционирования всех элементов государства и мирового сообщества. Постоянно растущий уровень конкурентности, стремительно увеличивающиеся потребности, практически неограниченное расширение сфер применения компьютерных продуктов и информационных технологий – предъявляют все время растущие требования к выпускнику средней школы. С целью повысить готовность выпускника в минимальные сроки (и даже желательно – на заключительном этапе обучения в школе) стать участником и инноватором государственной и мировой промышленности и экономики – создан курс "Компьютерные науки".

Для достижения учащимися массовой школы образовательных результатов на уровне требований современного мира социальных отношений, государственной жизни, деловой инициативы) необходимы:

- высокий уровень освоения учебных дисциплин, связанных с современными наукоемкими (высокими) технологиями;
- формирование учебных навыков владения методами и инструментами современных технологий, в первую очередь в компьютерной и информационной сфере – не только на уровне пользователя, но и на уровне креативного инноватора;

- проведение учебно-исследовательской и творческо-созидательной работы в технологических средах, отражающей современные реалии экономики и рынка;
- акцент на новых базовых компонентах содержания;
- применение методов работы и методов оценки работы, адекватных новым задачам и перспективам, а также ожидаемым результатам обучения.

Национальная экономика современной России, с постоянно увеличивающейся ролью и долей инновационных технологий, международный динамично развивающийся рынок информационных технологий – демонстрируют непрерывно растущую количественно и все более жесткую качественно потребность в IT-специалистах. При подготовке кадров для данной сферы необходима преемственность обучения на всех ступенях образования. Стремительно изменяющаяся информационная среда современного общества, научно-технический прогресс подстегивают сферу образования к столь же быстрой модернизации. Выпускник российской школы, решивший стать участником этого процесса, должен сегодня иметь знания сверх тех, дополнительные к тем, которые он получает в рамках базового курса информатики. Выпускник должен владеть практическими навыками работы с компьютерными и информационными технологиями, быть способным создавать программные и информационные продукты начального (но уже достаточно профессионального) уровня. Эффективное обучение в ВУЗе, успешное трудоустройство и карьерный рост напрямую зависят от степени профессиональных компетенций в сфере компьютерных и информационных технологий.

В 2007 году Ассоциация предприятий компьютерных и информационных технологий (АП КИТ) опубликовала документ «Профессиональные стандарты в области информационных технологий». В документе представлены должностные обязанности, профессиональные компетенции, требования к уровню образования и стажу для девяти наиболее массовых и востребованных профессий в IT-сфере:

1. программист;
2. системный архитектор;
3. специалист по информационным системам;
4. системный аналитик;
5. специалист по системному администрированию;
6. менеджер информационных технологий;
7. менеджер по продажам решений и сложных технических систем;
8. специалист по информационным ресурсам.
9. администратор баз данных.

Анализ этого документа однозначно свидетельствует о том, что сегодня национальная экономика России и международный рынок четко и совершенно однозначно определяют приоритет не только общих фундаментальных знаний выпускников школ и ВУЗов в области алгоритмизации и программировании, знании современных объектно-ориентированных языков и прикладного направления в подготовке будущих специалистов – но и требуют реальной подготовки выпускников в овладении принципиальными (базовыми) и, одновременно, практическими знаниями и навыками.

В существующих программах и УМК по информатике и ИКТ приоритетное внимание отдан освоению офисного пакета программ. Изучаются языки программирования, не предоставляющие прикладного выхода знаний учащихся. Не уделяется внимание самостоятельной проектной работе учащихся.

Данная учебная программа ориентирована на получение одновременно и фундаментальных знаний в области информатики и ИКТ, и на формирование практического навыка применения этих знаний в процессе выполнения учебных проектов в рамках реально используемых в экономике, промышленности и науке современных компьютерных и информационных технологий и подходов.

Место и роль курса в обучении

В настоящее время существуют проблемы, связанные с подготовкой старшеклассников к итоговой государственной аттестации в форме ЕГЭ. Большинство учебных программ по информатике и ИКТ ориентируются на изучение школьниками прикладных программ офисного пакета, на изучение философских и мировоззренческих проблем информационных технологий. Недостаточное внимание уделяется таким важным разделам информатики как теория программирования, основы теории информации, формирование практических навыков создания реальных программных и информационных продуктов, отражающих потребности рынка и науки. Без формирования у выпускника симбиоза глубоких знаний принципов алгоритмизации и программирования и практических креативных навыков создания реальных компьютерно-информационных продуктов – шансы его на успех в качестве эффективного члена и лидера в современном обществе резко снижаются.

Содержание курса дает учащимся и фундаментальное представление о современных подходах к путям и способам практической реализации изучаемой теории, а также формирует у него навыки практического применения этих представлений в рамках овладения современными компьютерно-информационными технологиями.

Содержание курса удовлетворяет требованиям доступности и понятности материала, его связи с практикой.

Основные цели и задачи курса:

1. формирование у школьников основ научного мировоззрения;
2. обеспечение преемственности между общим и профессиональным образованием;
3. создание условий для самореализации и самовоспитания личности;
4. формирование у школьников базового системного представления о теоретической базе современных компьютерно-информационных технологий;

5. формирование умения креативно и на практике находить эффективные решения исследовательских и практических задач;
6. формирование представления о взаимосвязи и взаимовлиянии современных компьютерных и информационных сфер с фундаментальными и прикладными науками.

Начальный уровень знаний, умений и навыков, необходимых для прохождения данного курса ученик должен **знать/понимать**:

- методы введения, обработки и вывода информации;
- требования к дружелюбности интерфейса современных компьютерно-информационных продуктов;
- основные свойства алгоритма, типы алгоритмических конструкций: следование, ветвление, цикл; понятие вспомогательного алгоритма – и пути их реализации в рамках доступных программно-информационных инструментов и продуктов;
- назначение и функции, используемых информационных и коммуникационных технологий;
- уметь:
- выполнять базовые операции над объектами: цепочками символов, числами, списками, деревьями; проверять свойства этих объектов;
- планировать и реализовывать эти операции в виде реального программного продукта;
- планировать, создавать и оперировать информационными объектами, используя графический интерфейс, адаптировать создаваемый продукт к потребностям потенциального клиента, учитывая требования дружелюбности системы "клиент – продукт";

Курс рассчитан на учащихся два семестра обучения в 10 и один семестр в 11 классах. Как вариант возможно изучение по блокам:

Вариант 1: 1 блок (начальный уровень – основы программирования на java) 34 часа в год 10 класс.

Вариант 2: 1-2 блок (начальный уровень, профильный уровень – объектно-ориентированное программирование и оконный интерфейс) 68 часов 10 класс или 34 часа 10 класс и 34 часа 11 класс

Вариант 3: 1-2-3 блок (полный курс- объектно-ориентированное программирование и программирование для ОС «Андроид») 68 часов 10 класс и 34 часа 11 класс.

Учебные могут проводиться как в рамках урочной, так и внеурочной деятельности. Предусматривается выполнение проектных работ.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН
10 класс (начальный уровень обучения)
2 часа в неделю в семестр
(вариант – 1 час в неделю в год)

№ темы	Тема	Количество часов
1	Основные моменты истории программирования. Классификация языков программирования. Установка ПО	1
2	Базовая структура программы на Java. Организация ввода и вывода. Запуск программы из терминала	1
3	Переменные, типы переменных. Объявление и инициализация переменных.	1
4	Решение задач к разделу 5	1
5	Блоки, область существования переменных	1
6	Арифметические операторы	1
7	Решение задач к разделу 7	1
8	Ветвление	1
9	Решение задач к разделу 8	2
10	Логические операторы	1
11	Решение задач к разделу 9	1
12	Цикл while	1
13	Решение задач к разделу 10	1
14	Цикл for	1
15	Решение задач к разделу 11	1
16	Класс Math	1

17	Решение задач к разделу 12	1
18	Массивы	1
19	Решение задач к разделу 13	2
20	Строковые и символьные переменные	1
21	Решение задач к разделу 14	1
22	Практикум по решению задач	4
23	Работа с файлами. Обработка исключений	1
24	Решение задач к разделу 17	1
25	«Продвинутые» системы разработки. Понятие jar-файла. Запуск файла из терминала. Импорт сторонних библиотек.	1
26	Практикум решения задач	4
Итого		34

СОДЕРЖАНИЕ КУРСА

1. Основные моменты истории программирования. Классификация языков программирования. Установка ПО. (1 час)

История языков программирования. Классификация языков программирования. Машинные языки. Машинные коды. Языки высокого уровня. Ассемблер. Языки Фортран, Паскаль, Бейсик, С. История возникновения языка Java. Развитие java, установка ПО. Понятие JDK и JRE .

2. Базовая структура программы на Java. Организация ввода и вывода. Запуск программы из терминала. (1 час)

Обязательные элементы программы (класса) на языке Java. Краткое объяснения каждого обязательного элемента. Допустимые изменения. Назначения каждой основной части класса. Главный метод. Организация консольного ввода и вывода. Значение комментариев и правила их написания. Редактор DrJava. Правила установки и запуска. Основные этапы создания класса. Написание

кода и возможные ошибки. Отладка кода и правила исправления ошибок. Сообщения об ошибках. Прогон класса и логические ошибки.

3. Переменные , типы переменных. Объявление и инициализация переменных. (1 час)

Смысл понятия "переменная". Цель использования переменных в классе. Переменные и значения (данные). Типы данные и типы переменных. Разнообразие типов данные и переменных. Правило соответствия типа данных и типа переменных – и наоборот. Объявление переменных. Возможные варианты объявления переменных. Инициализация переменных. Возможные варианты инициализации переменных. Объединение объявления переменных с инициализацией и отдельное выполнение объявления и инициализации. Возможные ошибки и их исправление.

4. Решение задач к разделу 5. (1 час)

5. Блоки, область существования переменных. (1 час)

Блоки в java. Вложенность блоков. Область существования переменных. Типичные ошибки, возникающие при неправильном применении блоков.

6. Арифметические операторы. (1 час)

Арифметические операторы в java. Приоритет операций. Изменение приоритета операций. «разбивка числа на цифры».

7. Вешение задач к разделу 7. (1 час)

8. Ветвление. (1 час)

Команда выбора в полной и неполной форме. Использование блоков. Логика выполнения команды выбора при неисполнении условия.

9. Решение задач к разделу 8. (2 часа)

10. Логические операторы. (1 час)

Переменные логического (булевого) типа и значения логического (булевого) типа. Значение "истина" и значение "ложь". Связь между

переменными логического типа и математическими выражениями. Использование переменных логического типа. Особенности использования переменных логического типа в командах вывода на экран.

11. Решение задач к разделу 9. (1 час)

12. Цикл while. (1 час)

Понятие цикла с "предустановленным" числом повторов и понятие "гибкого" цикла. Цикл while. Особенности этого цикла. Понятие "условия повтора". Анализ поставленной задачи с точки зрения выбора подходящего к решению типа цикла. Цикл while и использование логических переменных

13. Решение задач к разделу 10. (1 час)

Понятие блока. Правила написания блока в классе (программе). Использование блока в операторах выбора, правила написания таких блоков. Возможные ошибки и их исправление.

14. Цикл for. (1 час)

Понятие о цикле for. Структура цикла. Заголовок цикла и тело цикла. Правила написания заголовка цикла for. Понятие о переменной-счетчике повторов. Особенности описания и инициализации переменной-счетчика повторов. Варианты написания заголовка цикла for. Особенности написания тела цикла for. Возможные ошибки (синтаксические и логические) при использовании цикла for.

15. Решение задач к разделу 11. (1 час)

16. Класс Math. (1 час)

Необходимости использования математических и логических действий, которые не всегда можно выразить в виде стандартных выражений. Понятие системной библиотеки в языке Java. Системная библиотека Math. Правила использования этой библиотеки. Понятие случайного числа и смысл генерации случайных чисел. Правило

генерации случайного числа с использование системной библиотеки Math. Преимущества и недостатки использования генерации случайных чисел как замены вводу данных с клавиатуры.

17. Решение задач к разделу 12. (1 час)

18. Массивы. (1 час).

Понятие массива. Имя массива и индекс элемента. Типы массивов. Преимущества массивов для хранения больших данных.

19. Решение задач к разделу 13. (1 час)

Отработка навыков написания классов с использование системной библиотеки Math. Отработка навыков написания классов с использование случайных чисел. Возможные ошибки (синтаксические и логические) и их исправление.

20. . Строковые и символьные значения. (1 час)

Понятие строки и символа. Типы переменных и констант. Операции со строками и символами. Методы класса String. Числовое значение символа.

21. Решение задач к разделу 14. (1 час)

22. Практикум по решению задач. (4 часа)

В качестве заданий использовать тренировочные задания 20.2 ОГЭ по информатике разных лет.

23. Работа с файлами. Обработка исключений. (1 час)

Понятие файла. Хранение данных в файле. Дерево файлов и относительный путь к файлу. Классы FileReader и FileWriter. Методы этих классов. Обработка исключений.

24. Решение задач к разделу 17. (1 час)

25. Интегрированные системы разработки. (1 час)

Знакомство с другими системами разработки (на выбор учителя). Структура проекта. Система подсказок. jar – файл. Создание и запуск jar-файла.

26. Практикум по решению задач. (4 час)

В качестве заданий использовать тренировочные задания 27 ЕГЭ по информатике разных лет.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН 10 класс (профильный уровень обучения) 2 часа в неделю в семестр

№ темы	Тема	Количество часов
1	ООП – основные понятия	1
2	Использование методов в Java	2
3	Решение задач к разделу 2	1
4	Методы и массивы	1
5	Решение задач к разделу 3	1
6	Методы в других классах	1
7	Решение задач к разделу 4	1
8	Инкапсуляция	1
9	Наследование	1
10	Конструктор класса	1
11	Полиморфизм	1
12	Библиотеки импортируемых классов. Немного «расшифровки».	1
13	Библиотеки awt и swing	2
14	Решение задач к разделу 1.	1
15	Использование графики.	1
16	Решение задач к разделу 2	1

17	Обработка событий	1
18	Решение задач к разделу 3	1
19	Создаем диалог. Решение задач к разделу 4	1
20	Первый проект: Калькулятор	2
21	Работа с изображениями	1
22	Второй проект: Тестовая программа	2
23	Параллельные потоки	1
24	Третий проект: Игра	2
25	Итоговый проект	5
Итого		34

СОДЕРЖАНИЕ КУРСА

1.ООП – основные понятия. (1час)

Абстракция данных. Класс. Объект. Классификация подвидов ООП. Критика ООП.

2. Использование методов в java. (2 часа)

Дополнительные методы и их связь с главным методом. Понятие метода. Особенности выполнения класса, содержащего методы. Вызов метода (обращение к методу). Передача данных методу. Методы, возвращающие значение и методы, не возвращающие значений. Заголовок метода и тело метода. Правила написания заголовка метода. Команда return в методах, возвращающих значения. Типы методов. Классы с использованием методов и классы без использования методов – сравнение (преимущества и недостатки). Возможные ошибки (синтаксические и логические) и их устранение

3. Решение задач к разделу 2. (2 часа)

Отработка написания класса с использованием нескольких методов. Типы методов по принимаемым и возвращаемым значениям.

4. Методы и массивы. (1 час)

Методы принимающие и возвращающие массивы.

5. Решение задач к разделу 3. (1 час)

6. Методы в других классах. (1 час)

Написание методов в других классах. Вызов методов из других классов. Принцип создания файлов .class.

7.Решение задач к разделу 4. (1 час)

8. Инкапсуляция. (1 час)

Модификаторы доступа. Примеры и необходимость применения модификаторов доступа. Неявное задание модификаторов доступа.

9. Наследование. (1 час)

Создание дочерних классов с помощью extends. Наследование и переопределение методов. Создание дочерних классов с помощью new. Использование методов экземпляров класса.

10. Конструктор класса. (1 час)

Конструктор класса. Неявное и явное задание конструктора класса. Создание экземпляра класса с использованием конструктора класса.

11.Полиморфизм. (1 час)

Перегрузка методов. Перегрузка конструкторов класса. Достоинства полиморфизма.

12. Библиотеки импортируемых классов. (1 час)

Замещающие классы. Правила импорта. Часто используемые библиотеки java. «Расшифровка» ранее используемых конструкций.

13. Библиотеки awt и swing . (2 часа)

Фрейм. Панель. Виджеты. Разметка. Отработка навыков написания оконного интерфейса.

15.Использование графики. (1 час)

Класс Graphics. Методы класса. Класс Color. Цветовой круг.

16.Решение задач к разделу 2. (1 час)

17.Обработка событий. (1 час)

«Слушатели» в java.

18 Решение задач к разделу 3. (1 час)

19. Создаем диалог. Решение задач к разделу 4. (1 час)

20. Первый проект: Калькулятор. (2 часа)

21. Работа с изображениями. (1 час)

Графические файлы. Относительный путь к файлу. Размер файла. Метод drawImage.

22. Второй проект: Тестовая программа (2 часа)

23. Параллельные потоки. (1 час)

Класс Thread. Методы класса Thread. Создание экземпляра класса Thread. Запуск, остановка потоков.

24. Третий проект: Игра (2 часа)

25. Итоговый проект.(5 часов)

**УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН
11 класс (профильный уровень обучения)
2 часа в неделю в семестр**

№ (темы)	Название темы	Количество часов
1	История, развитие и перспективы ОС Android	1
2	Установка программного обеспечения. Виды IDE. Знакомство с редактором и правилами работы.	1
3	Проект. Установка приложения «Hello world» на Андроид устройство.	1
4	Структура проекта. Основные принципы языка	1

	разметки XML	
5	Файл разметки главной активности. Инструменты графического дизайна файла (WYSIWYG) и коды. Структура активности и свойства элементов.	1
6	Java-файлы проекта. Назначение, структура.	1
7	Работа с java-файлом главной активности. Действие onClick()	1
8	Программное размещение виджетов.	2
9	Основные виджеты и установка атрибутов	1
10	Проект «Калькулятор»	3
11	Использование ресурсов. Графика	1
12	Использование ресурсов. Стили.	1
13	Проект «Тестовая программа».	4
14	Создание собственного View	1
15	Использование координатной сетки.	1
16	Работа с файлами.	1
17	Организация нескольких потоков. Класс AsyncTask	1
18	Проект «Игра»	3
19	Использование основных виджетов	1
20	Итоговый проект	5
21	Защита проектов	2
Итого		34

Содержание курса

1. История, развитие и перспективы ОС Андроид. (1 час)

Разные виды ОС, степени их открытости, доступность для создания приложений. Развитие мобильных устройств. Особенности ОС Андроид. Безопасность при работе со сторонними приложениями.

2. Установка программного обеспечения. Знакомство с редактором и правилами работы. (1 час)

Различные IDE предназначенные для разработки приложений для ОС «Андроид» . Практическая установка редактора на персональный компьютер. Определение рабочей области. Создание нового проекта. Назначение окон редактора.

3. Проект. Структура проекта. Установка приложения «Hello world» на Андроид устройство. (1 час)

Структура проекта. Импорт, экспорт проекта. Сохранение проекта. Файл *.арк. Установка приложения в AVD и реальное устройство. Безопасность установки.

4. Основные принципы языка разметки XML. (1 час)

Язык разметки XML. Синтаксис языка. Возможные ошибки. Основные теги. Практическое сравнение графического расположения объекта и его разметка в xml файле.

5.Файл разметки главной активности. Инструменты графического дизайна файла (WYSIWYG) и коды. Структура активности и свойства элементов. (1 час)

Файл main.xml. Создание файла. Работа с xml-файлом в графическом режиме. Размещение различных объектов. Изменение свойств.

6. Java-файлы проекта. Назначение, структура. (1 час)

Расположение. Структура. «Обязательные» методы. Импорт классов.

7.Работа с java-файлом главной активности. Действие onClick(). (1 час)

Создание собственного метода «действие». Практическая работа по решению типовых задач.

8. Программное размещение виджетов. (2 часа)

Создание объекта с помощью **new**. Использование координат для размещения объектов. Изменение расположения объекта на разметке.

9.Основные виджеты и установка атрибутов. (1 час)

Разбор палитры объектов. Изменение свойств объектов с помощью панели «Свойства».

10.Проект «Калькулятор». (3 часа)

Создание приложения «Калькулятор». Разметка активности. Написание кода. Тестирование приложения на реальном устройстве.

11.Использование ресурсов. Графика. (1 час)

Подготовка графических объектов для использования в приложении. Вопросы масштабирования изображений. Различные способы использования графических изображений.

12.Использование ресурсов. Стили. (1 час)

Создание собственных стилей. Правила использования собственных стилей.

13.Проект «Тестовая программа». (4 час)

Подготовка ресурсов. Создание приложения «Тестовая программа». Тестирование приложения. Варианты дальнейшего развития приложения.

14.Создание собственного View . (1 час)

Повторение понятия «Переопределение метода». Класс Canvas. Создание векторных объектов.

15.Использование координатной сетки. (1 час)

Система координат разметки. Метод `setOnTouchListener()`. Применение приема считывания координат при касании экрана.

16.Работа с файлами. (1 час).

Практическая работа «Запись-чтение файлов».

17.Организация нескольких потоков. Класс AsyncTask. (1 час)

Создание и использование дополнительного потока. Класс AsyncTask и реализация методов этого класса.

18.Проект «Игра». (3 часа)

Создание сценария. Подготовка ресурсов. Написание кода. Тестирование приложения на реальном устройстве.

19.Использование основных виджетов. (1 час)

Практическая работа по программированию виджетов.

20.Итоговый проект. (5 часов)

Выбор темы. Подготовка ресурсов. Написание кода. Тестирование приложения

21.Защита проектов. (2 часа)

Результаты изучения учебного курса

Учащиеся должны знать/понимать/уметь:

1. Планировать, создавать и отлаживать программы на языке программирования Java.
2. Создавать программы в соответствии с поставленной общей задачей.
3. Выбирать соответствующие требованиям эффективного решения (создания программы) инструменты языка.
4. Интерпретировать результаты, получаемые в ходе исполнения реальных программ.
5. Оценивать сравнительную эффективность разных вариантов практического решения поставленной задачи (разных программ, достигающих сходного результата).
6. Использовать формальную логику для решения задач практического программирования.
7. Вычислять логическое значение сложного высказывания по известным значениям элементарных высказываний.
8. Структурировать решение, создавая программу на основе методов и подпрограмм.

Учащиеся должны использовать приобретенные знания и умения в практической деятельности и повседневной жизни:

1. Пользоваться справочными системами и другими источниками справочной информации. Оценивать достоверность информации.
2. Выбирать адекватные программные и аппаратные средства для решения поставленной задачи.
3. Использовать основные возможности языка программирования (Java) и используемого прикладного программного обеспечения – в соответствии с требованиями конкретной поставленной задачи.
4. Диагностировать ошибки программного обеспечения и устранять простейшие неисправности его работы.
5. Использовать системные библиотеки и уже созданные прежде программные компоненты для эффективного конструирования программного продукта для новой задачи.
6. Выполнять требования техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами и инструментами программирования