**Лабораторная робота №2**

Тема: Планирование учебного процесса по информатике

Цель: На основе анализа учебных программ по информатике и ИКТ сформировать умение составления тематического планирования и календарного плана учебной работы.

Календарно-тематическое планирование уроков информатики в 11 классе 198 часов (6 часов в неделю)

|  |  |  |
| --- | --- | --- |
| №  урока | Тема урока | Дата |
| 1 | Вступительный урок | 02.09.2020 |
| 2 | **Раздел 1.** Основы системного подхода  Понятие системы. Структура системы. Системный подход. Системный анализ. Модели систем. | 09.09.2020 |
| 3 | **Раздел 1.** Основы системного подхода  Информационные системы. Инфологическая модель предметной области. Систематизация. | 16.09.2020 |
| 4 | **Раздел 2.** Реляционные базы данных  Реляционные базы данных и СУБД. Проектирование реляционной модели данных. Создание базы данных. Простые запросы к базе данных. Параметрические запросы. Формы представления данных. Формы.  **Практическая работа № 1.** Создание базы данных.  **Практическая работа № 2.** Создание запросов.  **Практическая работа № 3**. Работа с формой. | 23.09.2020 |
| 5 | **Раздел 2.** Реляционные базы данных  Разработка структуры и создание многотабличной БД. Расширение базы данных. Составление сложных запросов Вычисляемые поля.  Основы языка SQL. Оператор IN. Вычитание множеств записей. Создание запросов на добавление, обновление и удаление данных. Сложные запросы к базе данных. Создания отчетов по одной и нескольким таблицам. Группировка данных в отчетах.  **Практическая работа № 4.** Создание многотабличной базы данных.  **Практическая работа № 5.** Создания отчетов по одной и нескольким таблицам.  **Лабораторная работа № 1.** Создание собственной базы данных | 30.09.2020 |
| 6 | **Контрольная работа №1 (Реляционные базы данных)** | 07.10.2020 |
| 7 | **Раздел 3.** Структуры данных  Понятия структур данных; простая переменная. Списки. Стек. Очередь. Деревья.  **Практическая работа № 6.** Структуры данных, создание стека. **Практическая работа № 7.** Работа с очередью | 14.10.2020 |
| 8 | **Раздел 4.** Основы теории графов  Графы. Поиски в ширину и в глубину Графы, способы задания: матрица смежности, список ребер. Графовые модели некоторых задач.  **Практическая работа № 8.** Основные понятия теории графов и способы представления графов.  **Практическая работа № 9.** Поиск в ширину и глубину. | 21.10.2020 |
| 9 | **Раздел 4.** Основы теории графов  Обходы графа. Графы и бинарные отношения. Алгоритм Дейкстры. Алгоритм Флойда – Уоршелла. Алгоритмы поиска остовов кратчайших маршрутов.  **Практическая работа № 10.** Нахождение минимального остова. **Лабораторная работа № 2.** Определение кратчайшего пути в графе | 28.10.2020 |
| 10 | **Раздел 5.** Основы динамического программирования  Динамическое программирование, табличный метод. Принцип ДП. Условия применимости динамического программирования. | 11.11.2020 |
| 11 | **Раздел 5.** Основы динамического программирования  Решение задач динамического программирования: самая длинная общая подпоследовательность, задача об оптимальной расстановке скобок, задача о нахождении оптимального пути между двумя пунктами, задача о рюкзаке.  **Лабораторная работа № 3.** Решение задач динамического программирования | 18.11.2020 |
| 12 | **Раздел 6.** «Жадные» алгоритмы  Понятие «жадный» алгоритм. Задача Прима-Краскала. Решение задач.  **Практическая работа № 11.** Решение задач с использованием «жадных» алгоритмов | 25.11.2020 |
| 13 | **Раздел 7.** Алгоритмы вычислительной геометрии  Векторное произведение, направление поворота; определение площади многоугольника; пересечение отрезков и определение положения точки относительно простого многоугольника;  **Практическая работа № 12.** Векторное произведение, направление поворота вектора и определение площади многоугольника.  **Практическая работа № 13.** Определение пересечения отрезков и определения положения точки относительно простого многоугольника. | 02.12.2020 |
| 14 | **Раздел 7.** Алгоритмы вычислительной геометрии  Построение выпуклой оболочки, определение пары ближайших и самых отдаленных точек.  **Лабораторная работа № 4.** Построение выпуклой оболочки | 09.12.2020 |
| 15 | **Контрольная работа №2 (**Основы динамического программирования - Алгоритмы вычислительной геометрии**)** | 16.12.2020 |
| 16 | **Раздел 8.** Мультимедийные презентации  Понятие шаблона презентации. Принципы стилевого оформления презентаций. Основные принципы дизайна слайдов. Создание элементов управления презентацией; настройка интерактивного оглавления с помощью гиперссылок; обеспечение возврата к оглавлению; добавление гиперссылок на документы Word; | 23.12.2020 |
| 17 | **Раздел 8.** Мультимедийные презентации  Добавление эффектов анимации; выбор эффектов анимации; настройка анимации; добавление управляющих кнопок на все слайды. Создание макросов.  **Практическая работа № 14.** Анимация в слайдовых презентациях.  **Лабораторная работа № 5.** Создание собственного проекта | 30.12.2020 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № | Темы уроков | Тип урока | Основное содержание | Вид контроля на уроке | Домашнее задание |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Контрольные вопросы

1. **Исторические предпосылки и становление школьной информатики. Приведите дату введения в средних школах предмета ОИВТ.**

В развитии отечественного школьного курса информатики выделяется несколько этапов (обычно – три), связанных со сменой парадигм преподавания курса и, соответственно, изменениями в методической системе обучения информатике. По нашему мнению, историю школьной информатики можно разделить на шесть этапов, соответствующих смене парадигм в школьном курсе информатики.

В рамках производственного обучения в школе и факультативных курсов возникло два направления обучения кибернетике и информатике в средней школе: общеобразовательное, связанное с изучением информационных процессов, принципов строения и функционирования самоуправляемых систем различной природы, автоматической обработкой информации (В.С. Леднев, А.А. Кузнецов: факультативный курс «Основы кибернетики» для 9-10 кл.) и прикладное в рамках дифференциации обучения в старших классах школы с производственным обучением, основанное на изучении программирования и устройства ЭВМ (В.М. Монахов, С.И. Шварцбурд и др.). Идея общеобразовательного курса получила признание и поддержку в лице ведущих специалистов того времени.

**1985 г.** характеризуется включением в учебные планы школ обязательного курса «Основы информатики и вычислительной техники» (в 1985 г.). Один из его идеологов – А.П. Ершов, который видел цель курса в обеспечении компьютерной грамотности школьников, под которой понималось умение программировать («Программирование– вторая грамотность», А.П. Ершов). Соответственно, основными понятиями курса были «компьютер», «исполнитель», «алгоритм», «программа». Для преподавания курса использовался первый школьный учебник по информатике , составленный авторским коллективом под руководством А.П. Ершова и В.М. Монахова.

***1985–1986 гг.:*** Разработка первого учебного пособия по информатике (А.П. Ершов, В.М.Монахов, А.А. Кузнецов, С.А. Бешенков, А.С. Лесневский, Э.И. Кузнецов, М.П. Лапчик и др.).

***1 сентября 1985 г.:*** Начало преподавания основ информатики и ВТ в массовой школе. Обучение информатике проходило под лозунгом, выдвинутым академиком А.П. Ершовым, «Программирование – вторая грамотность». Отечественная техника, выпускаемая в это время, имела программное обеспечение в основном для обучения программированию.

1. **Приведите определение информатики как науки. Когда она возникла и на какой основе?**

Информатика – это область человеческой деятельности, связанная с процессами преобразования информации с помощью компьютеров и их взаимодействием со средой применения.

Информатика появилась благодаря развитию компьютерной техники, базируется на ней и совершенно немыслима без нее. Кибернетика же развивается сама по себе, строя различные модели управления объектами, хотя и очень активно использует все достижения компьютерной техники. Кибернетика и информатика внешне очень похожие дисциплины и различаются, скорее всего, в расстановке акцентов:

· в информатике акцент делается на свойствах информации и аппаратно-программных средствах ее обработки;

· в кибернетике акцент делается на разработке концепций и построении моделей объектов с использованием, в частности, информационного подхода.

1. **Что является предметом и объектом информатики**

Объектом изучения научной информатики являются научная информация (логическая структура знания) и закономерности научных коммуникаций.

Предметом информатики как науки является информационная технология. Информационной технологией называется совокупность процессов обработки, представления и передачи данных.

1. **Цели и задачи обучения информатике в средней школе**

**Цели обучения информатике в школе:** формирование у учащихся представлений о свойствах информации, способах работы с ней, в частности с использованием компьютера.

**Задачи обучения информатике в школе:**

* познакомить школьников с основными свойствами информации, научить приемам организации информации и планирования деятельности, в частности учебной, при решении поставленных задач;
* дать первоначальные представления о компьютере и современных информационных и коммуникационных технологиях;
* дать представления о современном информационном обществе, информационной безопасности личности и государства.
* Содержание курса обучения регламентируется государственным образовательным стандартом по данному предмету.
* Анализ государственного стандарта, а также базовых нормативных документов, в частности примерного календарного планирования по предмету, показал, что в своем первоначальном виде курс ОИВТ, предлагаемый школам содержит в себе множество недостатков и не адаптирован к условиям непрерывного развития информационных технологий.
* Именно этот факт и послужил отправной точкой для разработки непрерывного курса обучения ОИВТ в школе (2-11 классы), апробация которого ведется с 2003-2004 учебного года. В настоящее время учителя информатики гимназии работают по данной программе.
* Программа в основном состоит из базового школьного курса ОИВТ и дополнена темами, содержащимися в вопросах вступительных экзаменов (тестов) по информатике в высших учебных заведениях.
* Преимуществом программы является ее четкая структурированность по основным разделам информатики и по годам обучения, что позволяет безболезненно варьировать содержание курса ОИВТ в зависимости от современного состояния развития информационных и телекоммуникационных технологий, и в то же время оставаясь в рамках требований госстандарта и нормативных методических положений. Структура программы показана на рисунке.
  + формирование умений использовать компьютер и программное обеспечение для решения практических задач.
* В соответствии с программой и требованиями госстандарта

1. **Перечислите основные компоненты алгоритмической культуры учащихся**

**1. Понятие алгоритма и его свойства.** Понятие алгоритма является центральным понятием алгоритмизации и, соответственно, основным компонентом алгоритмической культуры. В обучения алгоритмизации нет необходимости (да и возможности) использовать строгое математическое уточнение этого понятия, достаточно его толкования на интуитивно-наглядном уровне. Существенное значение при изложении приобретают такие содержательные свойства алгоритмов, как понятность, массовость, детерминированность и результативность.

**2. Понятие языка описания алгоритмов.** Задача описания алгоритма всегда предполагает наличие некоторого языка, на котором должно быть выполнено описание. По этой причине само понятие алгоритма находится в неразрывной связи с понятием языка как средства выражения (представления) алгоритма. Выбор языка в каждом отдельном случае определяется областью применения алгоритма, т.е., по существу, свойствами объекта (человека, автомата, компьютера), выступающего в роли исполнителя. Соблюдение требования строго следовать границам языковых возможностей в общении с тем или иным исполнителем служит в некотором роде первоосновой алгоритмизации.

**3. Уровень формализации описания.** Понятие уровня формализации описания неразрывно связано с понятием языка. Если описание составлено для автомата, то используемый при этом язык подчиняется строгим ограничениям, которые обычно могут быть сведены в систему формальных правил, образующих синтаксис языка. Сам язык в подобных случаях становится формализованным. Однако на практике в процессе разработки алгоритмов, особенно при построении предварительных описаний, могут использоваться языковые средства, не обязательно строго ограниченные. Более того, такая ситуация возможна и не только в процессе предварительной разработки. Если, к примеру, алгоритм адресуется человеку, то и окончательный вариант алгоритмизации может иметь неформальное, «расплывчатое» представление. Немалое множество используемых на практике алгоритмов «работают» именно в неформализованном варианте. Важно лишь, чтобы алгоритм был понятен исполнителю, т.е. не использовал средств представления, выходящих за границы его возможностей. Таким образом, применяемые на практике уровни формализации представления алгоритмов могут варьироваться в довольно широком диапазоне: от уровня полного отсутствия формализации до уровня формализации «в той или иной мере» и, наконец, до уровня «абсолютной» формализации.

**4. Принцип дискретности (пошаговости) описания.** Построение алгоритма предполагает выделение четкой целенаправленной последовательности допустимых элементарных действий, приводящих к требуемому результату. Организованная совокупность этих действий образует определенную дискретную структуру описания алгоритма, сообщающую ему ясность и четкость. В различных языках такие отдельные этапы алгоритма представляются различными средствами. В словесных представлениях алгоритма (на естественном языке) – это отдельные предложения, указания, пункты, в языке схем – это отдельные блоки, в объектном языке ЭВМ – это отдельные команды, в алгоритмическом языке высокого уровня – операторы.

**5. Принцип блочности**. Возможности языка, используемого для построения алгоритмов, вынуждают избирать ту или иную степень детализации описаний. Это обстоятельство не препятствует, однако, тому, чтобы в процессе работы по составлению требуемого алгоритма при описании его первоначальной схемы употребить язык, единицы действия которого более крупны по сравнению с возможностями исполнителя, которому алгоритм адресуется. По сути дела, речь в данном случае идет об умении расчленять сложную задачу на более простые компоненты. Такой путь приходится избирать всегда, когда задача оказывается достаточно сложной, чтобы алгоритм ее решения в нужном языке можно было описать сразу. В этом случае задача разбивается на информационно замкнутые части (блоки), которым придается самостоятельное значение, и после составления первоначальной схемы, связывающей части задачи, проводится работа по детализации отдельных блоков. Каждый из этих блоков может быть детализирован по только что описанному принципу. При окончательном построении алгоритма из блоков возможны два принципиально различных подхода:

а) детальное представление блока помещается в соответствующее место алгоритма, а сам блок, исчерпав свою роль общего приема поиска алгоритма, как бы «растворяется» в нем;

б) содержание блоков не встраивается в алгоритм, а в его соответствующих местах помещаются ссылки – обращение к размещенным отдельно блокам; окончательным алгоритмом считается совокупность главного алгоритма и всех его отдельных блоков (вспомогательных алгоритмов).

**6. Принцип ветвления**. Требование алгоритмической полноты языков, используемых для представления алгоритмов, должно обеспечивать наличие средств, позволяющих реализовывать в алгоритмических описаниях логические ситуации, т.е. ситуации, в которых требуется принятие решения в зависимости от заданных начальных условий. Организация таких алгоритмов требует умелого использования логических (разветвляющих) средств языка. Существенными компонентами алгоритмической грамотности здесь является осознание того, что:

а) описание должно предусматривать все возможные варианты исходных данных и для каждой их комбинации быть результативным;

б) для конкретных значений исходных данных исполнение алгоритма всегда проходит только по одному из возможных путей, определяемому конкретными условиями.

**7. Принцип цикличности.** Эффективность алгоритмических описаний в большинстве случаев определяется возможностью неоднократного использования одних и тех же фрагментов описаний при различных значениях входных величин. Именно на этом приеме основано построение описаний, не удлиняющихся при увеличении объема действий, предусматриваемых этими описаниями. Возвращение к повторному прохождению одного и того же фрагмента описания может быть организовано с применением логических средств языка, однако язык может содержать и специальные средства организации циклических алгоритмов (например, операторы цикла в языках высокого уровня). И в том и другом случае существенным компонентом алгоритмической культуры здесь является понимание общей схемы функционирования циклического процесса и, что особенно важно, умение выделять при построении алгоритмов повторяющуюся (рабочую) часть цикла.

**8. Выполнение (обоснование) алгоритма**. Существенно важным компонентом алгоритмической грамотности является постоянно привлекаемое в процессе алгоритмизации умение воспринимать и исполнять разрабатываемые фрагменты описания алгоритма отвлеченно от планируемых результатов – так, как они описаны, а не так, как может быть, в какой-то момент хотелось бы самому автору или исполнителю. Говоря иными словами, требуется развитое умение четко сопоставлять (и разделять) то, что задумано автором, с тем, к чему приводит фактически написанное.

**9. Организация данных.** Исходным материалом для алгоритма является информация или исходные данные, которые надлежит обработать. Составитель алгоритма обязан думать не только о том, как и в какой последовательности производить обработку, но и о том, где и как фиксировать промежуточные и окончательные результаты работы алгоритма.

1. **Перечислите компоненты, составляющие содержание компьютерной грамотности школьников**

Знание основных терминов и того, что они означают. Это помогает найти взаимопонимание с разными службами поддержки пользователей и вообще необходимо в ситуациях, когда обычное течение работы даёт сбой;

- Практические навыки работы с компьютером: умение подготовить компьютер к работе, включать и выключать его, владеть клавиатурой, уметь пользоваться текстовым и графическим редакторами, электронными таблицами;

- Понимание возможностей компьютера. Компьютер может быть лишь инструментом, с помощью которого выполняется работа. Он не заменяет самого человека;

- Некоторые технологические принципы работы интернета. Прежде всего то, что нельзя подключиться ни к какой сети передачи данных, не имея при этом специального устройства и линии связи;

- Применение и роль компьютеров в различных областях деятельности человека.

1. **Перечислите компоненты информационной культуры**

**Получение информации**– пользователь отбирает информацию рационально и эффективно.

**Оценивание информации** – пользователь оценивает информацию

критически и компетентно.

**Использование информации**– пользователь применяет информацию точно и творчески.

1. **Запишите 3-ёх этапную структура курса информатики и ИКТ, рекомендованную Министерством образования ДНР.**

Структура курса ИКТ делится на 3 этапа, каждый из этапов включает определенные классы школьного образования, такие как:

1. Начальное общее образование (3-4 класс)
2. Среднее общее образование (5-9 класс)
3. Высшее общее образование (10-11 класс)
4. **Проанализируйте программы основного общего образования и выпишите число недельных часов на изучение информатики в каждом классе**
5. Начальное общее образование (3-4 класс)

3 класс – 36 + 1

4 класс ­– 36 + 1

1. Среднее общее образование (5-9 класс)

5 класс – 33 + 2

6 класс ­– 33 + 2

7 класс – 33 + 2

8 класс ­– 33 + 2

9 класс ­– 33 + 2

1. Высшее общее образование (10-11 класс)

10 класс ­– 198

11 класс ­– 198