

Обучение программированию в базовом курсе информатики

- Цели обучения и место в базовом курсе
- Содержание обучения
- Методические подходы
- Требования к системе задач
- Этапы работы над задачей

Что такое программирование?

- **Теоретическая дисциплина**, обучающая искусству разработки программ для решения задач с помощью языков программирования (приемам и методам программирования)
«Программа = алгоритм + структура данных» (Н.Вирт)
- **Универсальная технология** решения различных задач:
 - алгоритмизация
 - кодирование
 - отладка
 - тестирование
- **Системная деятельность** по решению задач с использованием компьютера, требующая определенного уровня мышления и интеллекта

Элементы системной деятельности - общеобразовательное значение

- Написание программы решения задачи требует разработки **модели** её решения, учет различных условий протекания программируемого процесса – формируется системный подход к анализу процесса
- Поиск путей решения задачи в условиях ограничений (средства языка программирования, системы программирования, ресурсов компьютера) – развивает **гибкость мышления** и аналитические способности
- Запись алгоритма с помощью средств языка программирования – развивает навыки формализованного описания решения (формализации) и **формально-логическое мышление**
- Редко когда первый вариант программы оказывается идеальным. Многочисленные синтаксические, семантические и логические ошибки требуют организации системной деятельности по поиску и устранению ошибок – формируются **навыки рефлексивной деятельности** (компьютер помогает конкретизировать мышление)

Научение программированию изменяет способ, с помощью которого дети изучают учебные предметы, изменяет понятие задачи, подходы к решению задач (в чем это проявляется?)

Цели обучения программированию в базовом курсе информатики *(предметные)*

- расширение и углубление **представлений об автоматизированной обработке** информации (данных)
- знакомство с **приемами** разработки программ и **организации данных** в различных языках программирования
- знакомство с **технологией** решения задач с использованием систем программирования

Место линии программирования в базовом курсе информатики

Изучение программирования

- опирается на базовые знания и умения **алгоритмизации** (алгоритмы управления, основные алгоритмические конструкции)
- ГОТОВИТ К ПОНЯТИЮ **«информационная модель»** и процессу информационного и компьютерного моделирования

Место линии программирования и содержание обучения в базовом курсе

АЛГОРИТМИЗАЦИЯ

Задачи
управления
исполнителями
Алгоритмические
конструкции

ПРОГРАММИРОВАНИЕ

Задачи
обработки
информации
(данных) на
языках програм-
мирования

[Управление
информацией,
хранимой
в памяти
компьютера]

МОДЕЛИРОВАНИЕ И ФОРМАЛИЗАЦИЯ

???

Чему должны научиться?

(образовательные результаты)

- научиться выдвигать и обосновывать **идею решения** задачи (освоить методы программирования)
- научиться **структурировать** эту идею (модель данных, технология проектирования программы, структурный подход, пошаговая детализация и др.)
- научиться **формализовать** элементы полученной структуры средствами выбранного языка (правила представления данных, основные операторы и синтаксис языка, правила записи программы)
- научиться **анализировать** результат решения задачи при разных значениях исходных данных
- научиться **сравнивать** различные методы решения задачи (по эффективности, простоте, сложности алгоритма и т.п.)

Содержание обучения в базовом курсе информатики (min=12 час)

Из раздела алгоритмы:

Алгоритмы работы с величинами: типы данных ввод и вывод данных

Практические работы:

1. Разработка линейной программы.
2. Разработка программы, содержащей оператор ветвления.
3. Разработка программы, содержащей оператор цикла.
4. Разработка программы, содержащей подпрограмму.
5. Разработка программы по обработке одномерного массива.
6. Разработка программы, требующей использование логических операций.

Программирование:

Языки программирования, их классификация

Правила представления данных

Правила записи основных операторов: ввода, вывода, присваивания, ветвления цикла

Правила записи программ

Этапы разработки программы: алгоритмизация – кодирование – отладка – тестирование

Обрабатываемые объекты: цепочки символов, числа, списки, деревья, графы

Практикум:

Создание алгоритма (программы), решающего поставленную задачу (с использованием различных операторов, вспомогательных алгоритмов, обработки массивов)

Примерная программа изучения программирования в базовом курсе (Угринович Н.Д., Семенов А.Л.)

1. **Основные типы алгоритмических конструкций** (линейная, разветвляющаяся, циклическая) и их реализация в языке программирования.
2. **Линейный алгоритм.** Простые вычислительные алгоритмы. Понятие переменной, Имя, значение. Семантический смысл оператора.
3. **Разветвляющийся алгоритм.** Разветвляющиеся алгоритмы с символьными переменными. Программы контроля знаний по предметам.
4. **Графические возможности** языка программирования. Основные графические примитивы (точка, линия, прямоугольник, окружность).
5. **Циклические алгоритмы.** Понятие сценария. Циклические алгоритмы построения изображений. Реализация движения объектов на экране.
6. **Вложенные циклы.** Решение комбинаторных задач, игровые задачи.
6. **Вспомогательные алгоритмы.** Сценарий многоэлементных изображений и их реализация с помощью вспомогательных алгоритмов.
7. **Операции и Функции** работы с символьными переменными (склеивание, вырезка, длина и др.) Алгоритмы работы с текстом: поиск, замена, перевертыши, и др.
8. **Массивы.** Понятие таблицы и массива. Различные виды массивов и их описание. Способы заполнения и распечатки массивов.

Основы объектно-ориентированного визуального программирования -14 час (Угринович Н.Д.)

1. **Понятие алгоритма** свойства алгоритмов. Исполнители алгоритмов. Способы записи алгоритмов. Формальное исполнение алгоритмов

Примеры формализованной записи алгоритмов: «редактирование текста в TP», запись в виде макросов, и др.

2. Разработка проектов в системе программирования Visual Basic.
Объекты: свойства и методы

Работа с готовым проектом.
Изучение и изменение свойств объектов

3. Графический интерфейс проекта и событийные процедуры

Проект 1. «Форма и размещение на ней управляющих элементов»

4. Переменные: тип, имя значение.

Проект 2. «Переменные»
(объявление и присваивание)

5. Арифметические, строковые и логические выражения.

Проект 3. «Калькулятор»

Основы объектно-ориентированного визуального программирования -14 час (Угринович Н.Д.)

5. Математические функции

Проект «Инженерный калькулятор», «Треугольник»

6. Строковые функции

Проект «Строковый калькулятор»
Проект «Регистрация»

7. Функции ввода-вывода

Проект «День рождения»

8. Функции даты и времени

Проект «Тест» + Регистрация»

9. Структура ветвления

Проект «Тест» + «Отметка»

10. Структура «выбор»

Проект «коды символов»

11. Структура «цикл»

Проект «Графический редактор»

12. Графические возможности языка

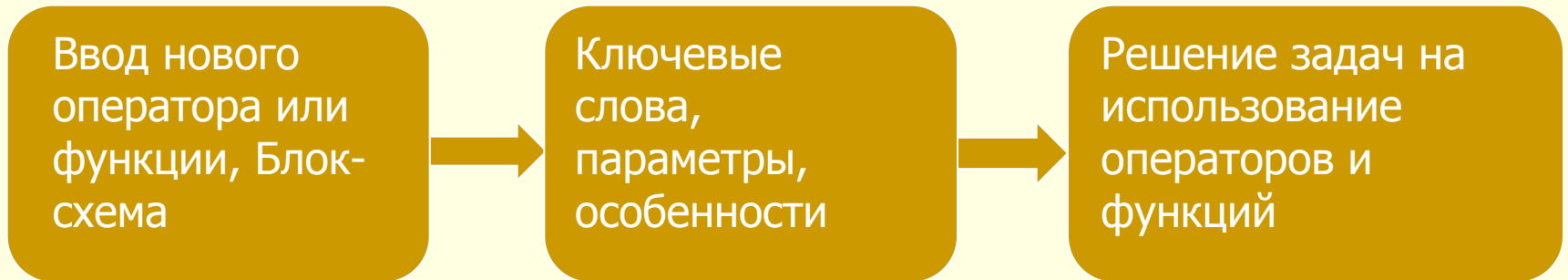
-

13. Анимация в языке

Творческий проект «Анимация»

Традиционная методика обучению программированию

Изучение особенностей использования основных конструкций конкретного языка программирования:



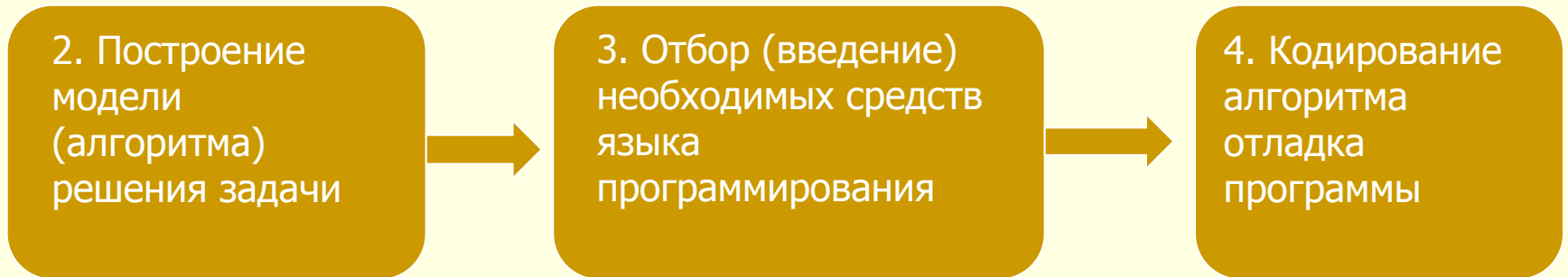
Логическая, методически грамотная последовательность с точки зрения принципов научности, понятности и доступности

Происходит подмена целей обучения: обучить языку программирования (кодированию на формальном языке) – вместо формирования общих подходов к решению задач на компьютере

Содержание обучения **изолировано** от основного содержания курса информатики

Отзадачный подход к обучению программированию

1. Изучение условия задачи (исходные данные – результат, зависимость результата от исходных данных, проигрывание возможных ситуаций и т.д.)



5. Анализ полученного результата, соответствие условию задачи

Основные идеи программирования и конструкции языка вводятся в процессе решения задачи, затем обобщаются и становятся объектом изучения

Главная цель – обучение решению задач с использованием средств языка программирования, освоение технологии программирования

Основные требования к системе задач

1. Задача **для объяснения** нового материала должна быть неформализованной, практически значимой и интересной! При ее решении потребуются новые приемы, операторы, функции... (расширять знания о методах решения и языке программирования)
 - новая конструкция языка вводится в том объеме, который необходим для решения данной задачи (изучение операторов - фоновая задача)
2. Задачи **для закрепления** должны предусматривать использование изученных операторов в модифицированном виде, отражать новые аспекты его применения (цепочка задач или задача с продолжением)
 - показывается что данная конструкция допускает видоизменения
 - после решения цепочки задач вводится строгий формат конструкции
3. Задачи **для изучения типовых методов** решения изначально формулируются «нечетко» и доопределяются в процессе диалога с учащимися (примеры: мишень, ср. рост, таблица умножения)
4. Задачи **для самостоятельного** решения (закрепления и проверки изученного) должны быть разного уровня - репродуктивного, продуктивного и с элементами творчества

Основные этапы работы над задачей

1. Знакомство с текстом и обсуждение (выявление проблемы, выявление ассоциаций в практической деятельности)
2. Обсуждение различных подходов к решению задачи, разбиение на подзадачи, выбор методов решения, формализованное описание действий в соответствии с возможностью компьютера и языка программирования
3. Построение первого варианта решения - частично формализованного и не обязательно верного (на естественном или алгоритмическом языке)

Основные этапы работы над задачей

4. Определение какие данные (переменные, константы, какого типа) потребуются. Введение обозначений.
Описание основных и промежуточных переменных
5. Запись на языке программирования фрагментов решения (наиболее трудные - обсуждаются фронтально учителем, остальные - учащиеся)
6. Сборка программы в целом
7. Ручная отладка программы - проигрывание различных ситуаций, отлавливание явных ошибок
8. Компьютерное тестирование программы (составить тесты - наборы исходных данных). Анализ результатов и коррекция
9. Обсуждение возможностей применимости и модификации

Итоги

(уточнение целей)

Изучение программирования остается **«классикой»** информатики

Однако, основным является не программа как таковая, а **задача**, которая решается с компьютером этой программой.

И здесь основным является модельный подход, поскольку алгоритм и программа - это **модели деятельности** субъекта по решению задачи.

Язык программирования и система программирования – это **средства** общения с компьютером (исполнителем), которому мы поручаем решение задачи (**средства автоматизации информационного процесса обработки информации**)

Итоги

(образовательный потенциал)

В процессе обучения программированию удается организовать **поисковую, исследовательскую деятельность** по решению проблем. Учим формулировать и проверять гипотезы

Углубляются представления о возможностях компьютера в автоматизированной обработке и хранении информации

Актуализируется значимость **человеческого интеллекта**: программа настолько хороша, насколько умён её создатель. В основе эффективного функционирования компьютера - интеллект человека

Программирование - островок **развивающего обучения**, который может дать не частные, а глубокие общие результаты в образовательном и воспитательном плане

Задание 1

(в портфель второго семестра)

Составить систему разноуровневых задач для организации практикума по одной из тем раздела «Программирование» базового курса информатики

Контрольные вопросы:

(ответы отправлять по адресу yakovleva@kspu.ru

имя файла: **тест2 фамилия.doc** тема: **4 курс филиал Ачинск**)

1. Укажите не менее трех причин, почему следует изучать программирование в базовом курсе информатики
2. Выявите общеучебные виды деятельности учащихся, развитие которых возможно в процессе решения задач по программированию
3. В чем вы видите методические сложности освоения программирования в базовом курсе информатики
4. Предложите оригинальный методический подход к освоению любой единицы содержания раздела «Программирование». Свой выбор обоснуйте

Основные вопросы курса ТиМОИ, изучаемые в 8 семестре

- Обучение программированию в базовом курсе информатики
- Моделирование и формализация как системообразующая линия курса информатики
- Вопросы социальной информатики в ШКИ
- Концепция непрерывного курса информатики в общеобразовательной школе
- Место школьного курса информатики в ИОС школе
- Государственный образовательный стандарт по информатике и ИКТ

Содержание портфеля достижений студента по ТиМОИ за 8 семестр

- Практикум по программированию (система разноуровневых задач)
- Методическое планирование системы уроков по телекоммуникационным технологиям
- Проект «Методическая система обучения в рамках отдельной темы школьного курса информатики для основной школы»