


Использование технологии проблемного обучения на уроках информатики

Презентация опыта работы
учителя МБОУ СОШ №25

Карповой М.В.



Хоть выйди ты не в белый свет,
А в поле за околицей, —
Пока идешь за кем-то вслед,
Дорога не запомнится.
Зато, куда б ты ни попал
И по какой распутице,
Дорога та, что сам искал,
Вовек не позабудется.

(Николай Рыленков)

Проблемное обучение

Проблемное обучение - это такая организация учебных занятий, которая предполагает создание под руководством преподавателя проблемных ситуаций и активную самостоятельную деятельность учащихся по их разрешению, в результате чего и происходит творческое овладение профессиональными знаниями, навыками и умениями и развитие мыслительных способностей

(Г.К. Селевко, 1998).

Проблемное обучение — это совокупность таких действий, как организация проблемных ситуаций, формулирование проблем, оказание ученикам необходимой помощи в решении проблем, проверка этих решений и, наконец, руководство процессом систематизации и закрепления приобретенных знаний

(Оконь В. Введение в общую дидактику. М.: Высшая школа, 1990, 383 с).

Структура проблемного урока



Классификация проблемных ситуаций



Примеры проблемных ситуаций на уроках

Изучение темы «Ветвление» на языке Паскаль»

(ситуация «не могу»)

Урок комбинированный. Задается задача: найти корни квадратного уравнения $a \cdot x^2 \pm b \cdot x \pm c = 0$. Условный оператор ученикам пока незнаком. Ученики решают ее используя операторы линейного алгоритма: находят дискриминант, корни по формулам и выводят результат $d = b^2 - 4 \cdot a \cdot c$ $x_1, x_2 = (-b \pm \sqrt{d}) / 2 \cdot a$. Для значений $a=2, b=10, c=5$ находятся корни, а для значений $a=5, b=2, c=10$ компьютер не находит решение. Вопрос. Почему я не могу найти корни?

Возникает проблемная ситуация. В результате поставленного задания знание предстает перед ним, как **требуемое** неизвестное **знание**.

После этого я приступаю к объяснению нового материала, затем учащиеся успешно решают поставленную задачу. Теперь изучение операторов языка Паскаль для ученика открытие нового, что соответствует ФГОС.

Примеры проблемных ситуаций на уроках

Изучение темы «Решение логических задач»

(ситуации «несоответствие», «конфликт», «неопределенность»)

Урок комбинированный. Предлагается задача

При составлении расписания на понедельник в IX классе преподаватели высказали просьбу завучу.

Учитель математики: **«Желаю иметь первый или второй урок».**

Учитель истории: **«Желаю иметь первый или третий урок».**

Учитель литературы: **«Желаю иметь второй или третий урок».**

Какое расписание будет составлено, если по каждому предмету может быть только один урок?

На предыдущих уроках рассмотрены основные операции алгебры логики и формулы преобразования логических выражений.

Возникает проблемная ситуация, когда надо применять приобретённые знания в новых условиях, т.к. данная задача решается методом рассуждений. Предлагаю ученикам самостоятельно решить ее. После обсуждения вместе разбираем задачу, обобщая все высказывания

Примеры проблемных ситуаций на уроках

Изучение темы «Решение логических задач»

(ситуации «несоответствие», «конфликт», «неопределенность»)

- Пусть в просьбе математика первое высказывание истинно, а второе – ложно.
«Желаю иметь первый или второй урок».

1 0

Т.е. первым будет урок математики.

- Тогда в просьбе учителя истории первое высказывание ложно, а второе истинно, т.е. третьим будет урок истории. «Желаю иметь первый или третий урок».

0 1

- Значит, в пожелании учителя литературы окажется истинной первая часть, т.е. урок литературы будет вторым.

«Желаю иметь второй или третий урок».

1 0

Итак: I урок – математика,

II урок – литература,

III урок – история.

Далее предлагаются задачи, которые сами по себе уже задают проблемный вопрос и разбираются на их примере другие методы решения: табличный и средствами алгебры логики. Создание проблемной ситуации (несоответствие, конфликт, неопределенность) позволяет решать нетиповые задачи, находить рациональные способы их решения, творчески подходить к учебной деятельности, активно, с интересом участвовать в собственном учебном процессе.

Фрагменты урока с использованием проблемных ситуаций

Изучение темы «Одномерные массивы. Обработка одномерных массивов»

(ситуации «не могу», «неопределенность»)

Создание проблемной ситуации

Программист за 10 мин должен обработать показания 1000 датчиков в центре управления полетами. Значит надо задать 1000 переменных. За 10 минут он не успевает это сделать. Как быть? Задача невыполнима на данном этапе (ситуация не могу и неопределенность)

Выслушиваю предложения учеников и подвожу их к мысли, что нужны новые знания - это работа с массивами.

Вводится понятие массива

Обращаю внимание учеников на то, что массив позволяет быстро работать с большими объемами данных

Массив – это совокупность конечного числа данных одного типа.

Например, последовательности чисел, таблицы, списки.

В программировании массив – это последовательность однотипных элементов, имеющих общее имя, причем каждый элемент этой последовательности определяется порядковым номером (индексом) элемента. Применяется для обозначения объектов, аналогичных числовым последовательностям в математике. $(a_1, a_2, a_3, \dots, a_n)$

Фрагменты урока с использованием проблемных ситуаций

Постановка учебной задачи

Начнем сегодняшний урок с решения задачи.

С клавиатуры вводится n чисел (числа могут повторяться). Необходимо подсчитать количество чисел равных наименьшему числу.

Давайте составим алгоритм решения задачи.

Какой выход из сложившейся ситуации?

Или записывать все вводимые числа на листочек (при небольших значениях n), или нужно где-то сохранять вводимые числа.

И так, ребята мы подошли к тому, что для решения задачи вам нужно пополнить багаж знаний. Запишите тему нашего урока: «Одномерные массивы. Обработка одномерных массивов

Планируемый результат: Ученики начинают разрабатывать алгоритм. Ввести n (количество вводимых чисел). Найти наименьшее число.

Ввести число и запомнить его в переменную min .

В цикле пока не просмотрим n чисел делать:

Ввести число и запомнить его в переменную x

Сравнивать x с min . Если $x < min$, то записываем в переменную min значение x . Сравнить введенные ранее числа со значением min .

А как? Ведь мы не запоминали вводимые числа. Возникла проблемная ситуация.

Фрагменты урока с использованием проблемных ситуаций

Поиск решения

Итак, для чего нужны массивы и когда и как мы их используем?

Резюмирую: если работа программы связана с хранением и обработкой большого количества однотипных переменных, для их представления в программе можно использовать массивы.

Планируемый результат: Ученики начинают формулировать задачи для решения проблемы:

Узнать, что такое массив?

Что можно хранить в массиве?

Для чего он используется?

Как записать в массив число?

Как считать число из массива?

Какие действия можно выполнять над информацией, помещенной в массив?

(слайды 13, 14)

Структура массива всегда однородна. Массив может состоять из элементов типа *integer* , *real* или *char* , либо других однотипных элементов.

Программа может сразу получить нужный ей элемент по его порядковому номеру (индексу).

Номер элемента массива называется **индексом**. Индекс – это значение порядкового типа, определенного, как **тип индекса** данного массива.

Описание одномерных массивов

```
Var A: Array [1..1000] Of real;  
i:integer;
```

A – имя переменной;

Array – служебное слово (в переводе с английского означает «массив»);

[1..1000] – количество элементов;

Of – служебное слово (в переводе с английского означает «из»).

Тип индекса – любой порядковый тип, кроме типов *integer*, *longint*.

Тип же самих элементов может быть любым, кроме *файлового* типа.

Фрагменты урока с использованием проблемных ситуаций

Выражения решения

На этом этапе идет объяснение нового материала с использованием слайдов презентации. Показываю приемы работы с массивами. При этом возникают **новые проблемные вопросы** для каждой задачи, например: *Как вы думаете, при выполнении программы обязательно заполнять все ячейки данными? Почему? Если ячейка не заполнена то, какое значение в ней находится? Может ли реальное количество элементов в массиве может быть меньше, чем указано при описании?*

Планируемый результат:
Заслушиваются ответы учеников. Обсуждаем и делаем выводы. При выполнении программы вовсе не обязательно заполнять все ячейки данными (в этом случае значение ячейки будет равно нулю), т. е. реальное количество элементов в массиве может быть меньше, чем указано при описании, но ни в коем случае не должно быть больше. Решаем поставленную задачу используя алгоритм нахождения минимума.
(слайды 16-21)

Ввод одномерных массивов

```
Var i: integer;  
    X: array [1..30] of Integer;  
Begin  
    For i := 1 To 30 Do Read (X[i]);  
  
End.
```

Массив **X** вводится с помощью цикла с параметром, где в качестве параметра используется *индекс* элемента массива (*i*).

Заполнение одномерного массива случайными числами

```
var a: array [1..10] of Integer;  
i: integer;  
begin  
  randomize;  
  for i:=1 to 10 do  
    a[i]:=random(200)-56;  
end.
```

Вывод одномерных массивов в Паскале

Вывод одномерного массива осуществляется также поэлементно.

Для вывода массива можно использовать любой цикл.

Например:

- 1) `For i := 1 To n Do Write (x[i], ' ')`
- 2) `For i := 1 To n Do Write (x[i]:4)`
- 3) `For i := 1 To n Do Writeln (x[i])`
- 4) `While i<=n do Begin Write (x[i]:4); i:=i+1; End.`
- 5) `Repeat Write (x[i]:4); i:=i+1; Until i=n;`

Сумма, произведение элементов

```
...  
s:=0; p:=1;  
for i:=1 to 10 do  
  begin  
    s:=s+a[i]; p:=p*a[i];  
  end;  
...
```

Выбор по условию

...

k:=0; s:=0; p:=1;

for i:=1 to 10 do

 if {условие} then

 begin

 k:=k+1; s:=s +a[i] ;

 p:=p*a[i] ;

 end;

...

Максимальный, минимальный элемент

...

```
max:= a[1]; min:= a[1];
```

```
for i:=1 to n do
```

```
  begin
```

```
    if a[i] > max then begin max:= a[i]; nmax:=i;
```

```
  end;
```

```
    if a[i] < min then min:= a[i];
```

```
  end;
```

...

Фрагменты урока с использованием проблемных ситуаций

Реализация продукта

Подведение итога урока. А теперь, ребята вернемся к целям урока. Скажите на все вопросы мы нашли ответы? Чем ценны массивы? Каким образом задается описание массива, что в нем указывается? Каким образом задается обращение к элементу массива? Почему при описании массива предпочтительнее употреблять константы, а не указывать размеры массива в явном виде? Сможем мы быстро решить задачу обработки данных в ЦУП?

Ученики. (Предполагаемый ответ)
Массивы ценны тем, что: Индексы элементов массива обеспечивают доступ не к одному, а к последовательности элементов. Обработка массивов производится при изменении индексов элементов. Сразу можно хранить и обрабатывать большое количество однотипных данных, значит мы сможем быстро решить поставленную задачу.



**СПАСИБО ЗА
ВНИМАНИЕ!**

Использованные источники и литература

1. Информатика и ИКТ: учебник для 9 класса. / И. Г.Семакин, Л.А.Залогова, С.В.Русаков, Л.В.Шестакова. - 3-е изд., испр. – М.: БИНОМ. Лаборатория знаний, 2009.
2. Рапаков Г.Г., Turbo Pascal для студентов и школьников/ Г.Г. Рапаков, С.Ю. Ржеуцкая. Спб.: БХВ-Петербург, 2007
3. Окулов С.М., Основы программирования/ С. М. Окулов. – М.: БИНОМ. Лаборатория знаний, 2004.