**Тема урока: Вспомогательные алгоритмы. Метод последовательной детализации и сборочный метод**

**Цели:**

Знакомство с новыми понятиями: вспомогательный алгоритм, метод последовательной детализации алгоритма, формирование навыка работы со вспомогательными алгоритмами; закрепление навыков по составления линейных алгоритмов,

**Задачи:**

* *учебная* – изучить понятия: вспомогательный алгоритм детализация алгоритма, формирование навыков составления основных и вспомогательных алгоритмов;
* *развивающая* – развитие алгоритмического мышления, памяти, внимания, логического мышления, познавательного интереса, коммуникативной культуры, учебно-познавательной компетенции;
* *воспитательная* – развитие познавательного интереса, способствовать воспитанию в детях милосердия, ответственности, взаимопонимания, взаимоуважения, взаимопомощи и поддержки.

**Тип урока**: урок закрепления знаний и изучение нового материала.

**Вид урока**: комбинированный урок (лекция и практика).

**Оборудование и программное обеспечение:** презентации PowerPoint**.**

**Структура урока:**

1. Организационный момент

II. Актуализация знаний

Групповая форма работы.

III. Теоретическая часть

Постановка проблемной ситуации. Объяснение с помощью презентации.

IV. Закрепление знаний Практическое задание.

V. Итог урока Рефлексия.

VI. Домашнее задание

**ХОД УРОКА**

**I. Организационный момент**

Приветствие. Проверка присутствующих. Установление психологического и эмоционального контакта с детьми. Проверка домашнего задания.

**ІІ. Актуализация опорных знаний**

**Учитель**: мы продолжаем работать с учебным исполнителем «Черепаха». И прежде чем мы будем решать новые задачи, давайте вспомним, что мы проходили на прошлых уроках. Для этого я вам раздам листы, на которых нарисованы ребусы. Ваша задача в парах разгадать свой ребус и написать определение к получившемуся слову.

*Выполнение 2 минуты.*

**Учитель**: Давайте проверим, что получилось. (Учащиеся говорят свои ответы, учитель параллельно демонстрирует на экране ребусы).

*Черепаха – графический учебный исполнитель.*

*Алгоритм – понятное и точное предписание исполнителю выполнить конечную последовательность команд, приводящую от исходных данных к искомому результату.*

*Точность. Свойство алгоритма.*

*Точность алгоритма означает, что каждая команда должна пониматься однозначно.*

*Какие свойства алгоритмов вы знаете еще (ответы учащихся)*

*Программа – это алгоритм, записанный на языке исполнителя.*

**II. Мотивация, целеполагание, постановка проблемы**

Давайте попробуем представить себе, как мы с вами совершим поход к реке. Нам предстоит переплыть реку на лодке. Необходим алгоритм переправы через реку, учитывая, что в лодку помещаются только два человека. Давайте попробуем его составить.

*Вот ты (ученик) проговаривай команд, а я буду показывать вам на доске.*

|  |  |
| --- | --- |
|  | **Алгоритм:**   1. Сесть в лодку двум туристам 2. Переплыть 3. Высадить одного туриста 4. Переплыть 5. Сесть в лодку одному туристу 6. Переплыть 7. Высадить одного туриста 8. Переплыть 9. Сесть в лодку одному туристу 10. Переплыть 11. Высадиться всем туристам |
|  | Какую особенность в этом алгоритме вы можете отметить? |

( *Учащиеся отмечают повторение команд 2-9*.)

Давайте подумаем, какой недостаток этого алгоритма.

(*Потрачено много времени на написание повторяющихся команд*)

Очень много времени мы потеряли на запись повторяющихся команд. Как же можно сократить запись данного алгоритма? Ответить мы сможем на данный вопрос после рассмотрения сегодняшней темы урока:

(*Учащиеся записывают в тетради число и тему урока*)

«Вспомогательные алгоритмы». Итак, кто сможет сказать, о чем будет идти речь на сегодняшнем уроке (ответы учащихся)

Сегодня мы узнаем, какие алгоритмы называются вспомогательными и как оформляются такие алгоритмические структуры.

**III. Объяснение нового материала.**

Давайте повторяющиеся действия выведем в отдельный алгоритм, который назовем – Переправа. Посмотрите как будет выглядеть наш алгоритм:

|  |  |
| --- | --- |
|  | **Алгоритм:**   1. Сесть в лодку двум туристам 2. **Выполнить алгоритм Переправа** 3. **Выполнить алгоритм Переправа** 4. Переплыть 5. Высадиться всем туристам   Какие изменения произошли с нашим алгоритмом ? (он стал короче) |

Алгоритм, который мы получили будет называться **основным**, а алгоритм Переправа будет являться **вспомогательным.**

|  |  |
| --- | --- |
| **Алгоритм Переправа** | **Алгоритм Основной** |
| 1. Переплыть 2. Высадить одного туриста 3. Переплыть 4. Сесть в лодку одному туристу | 1. Сесть в лодку двум туристам 2. **Выполнить алгоритм Переправа** 3. **Выполнить алгоритм Переправа** 4. Переплыть 5. Высадиться всем туристам |

*Учащиеся записывают в тетради*

Вспомогательный алгоритм — это алгоритм решения некоторой подзадачи из исходной (основной) задачи.

Вспомогательный алгоритм, записанный на языке программирования, называется подпрограммой или процедурой

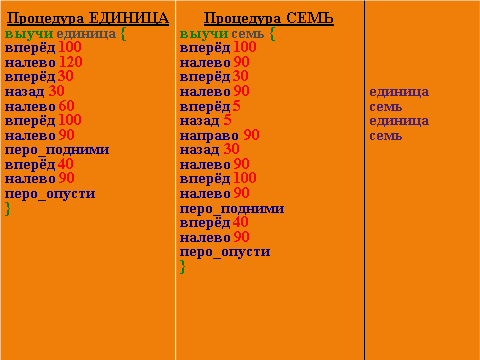
Вспомогательный алгоритм – алгоритм, снабженный таким заголовком, который позволяет вызвать его из других алгоритмов (должен иметь свое имя).

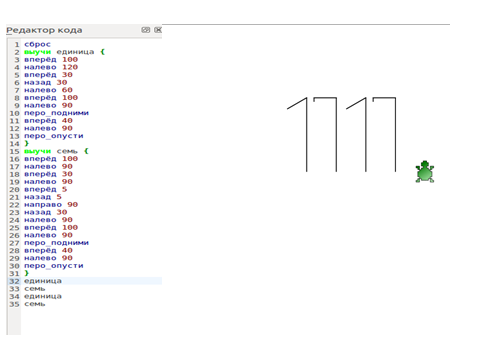
Из основной подпрограммы происходит обращение к вспомогательному алгоритму по его имени.

**IV. Закрепление**

Составим алгоритм написания числа 1717 для исполнителя «Черепашка». В языке программирования Лого ключевым словом, обозначающим начало подпрограммы, является, **выучи.**

Учащиеся за компьютерами пишут подпрограммы для число один и семь, и сверяют написанное с программой на экране.





Дальше записывают главную программу

Здесь главная программа - это команда **сброс** и все что идет после закрывающей фигурной скобки. От команды **выучи** и до закрывающей скобки включительно — это подпрограмма.

Метод программирования, при котором сначала составляется множество подпрограмм, которые могут понадобиться при решении задачи, а затем пишется основная программа, содержащая обращения к ним, называется ***сборочным методом***, или ***программированием снизу вверх***.

Подпрограммы могут быть объединены в ***библиотеку подпрограмм*** и сохранены в долговременной памяти компьютера. Такую библиотеку можно постепенно пополнять новыми подпрограммами.

Метод программирования, при котором сначала записывается основной алгоритм, а затем описываются использованные в нем вспомогательные алгоритмы, называется методом ***последовательной детализации***, или ***программированием сверху вниз***.

**Физминутка**

Давайте немного отдохнём.

Упражнения для рук и плечевого пояса:

1. Поднять плечи, опустить плечи. Повторить 6 – 8 раз. Расслабить плечи.

2. Руки согнуть перед грудью. На счет 1 – 2 – пружинящие рывки назад согнутыми руками, на счет 3 – 4 – то же, но прямыми. Повторить 4 – 6 раз. Расслабить плечи.

Упражнения для туловища и ног:

1. На счет 1-2-шаг влево, руки к плечам, прогнуться. На счет 3 – 4 – то же, но в другую сторону. Повторить 3 - 4 раза.

2. Ноги врозь, руки за голову. На счет 1 – резкий поворот налево, на счет 2 – направо. Повторить 3 - 4 раза.

Гимнастика для глаз:

1. Сядьте на стул, закройте глаза, расслабьте мышцы лица, свободно, без напряжения откиньтесь на спинку стула, положите руки на бедра (10 – 15 секунд).

2. Откройте глаза и посмотрите вдаль перед собой (2 – 3 секунды). Переведите взгляд на кончик НОСА (3 – 5 секунд). Повторите 2 раза.

**Практическое задание**

Написать с использованием процедур программу вывода на экран число 333

* 1. **Итог урока**

1. Какие вопросы были поставлены перед нами?

2. Получены ли ответы на поставленные вопросы?

*Рефлексия*

**Облако "тегов",** которые необходимо дополнить. сегодня я узнал...

* + было трудно…
  + я понял, что…
  + я научился…
  + я смог…
  + было интересно узнать, что…

Каждый ученик выбирает по 1-2 предложения и заканчивает их.

Оценивание работы учеников на уроке.

* 1. **Домашнее задание**

Выучить конспект.

параграф 5, № 7 стр.32 (рисунки1,2).