**Разработка вариантов опорных листов по усвоению базовых понятий учебного раздела базового курса информатики**

**Назначение**: Опорные листы необходимы для быстрого освоения учебного материала, для подготовки к экзамену или контрольной.

**Преимущество**: Применение опорных листов на занятиях позволяет упростить обработку материала, способствует обучению студентов структурировать материал, а также проявлять творческий подход к обучению. Поэтому их использование актуально в наши дни.

**План конструирования**:

1. Название темы
2. Выбор основного в изученном материале.
3. Определение ключевых слов и понятий, которые отражают суть темы.
4. Выделение подтем.
5. Выделение основных условных обозначений.
6. Построение схемы с обозначением структуры будущего плана.
7. Разделение материала на блоки.
8. Оформление полученного конспекта с помощью цветных маркеров, ручек, текстовыделителей. Требуется подчеркнуть главное, поставить знаки вопроса или восклицания возле спорных или важных моментов.
9. При необходимости обозначить вопросы, которые требуют дальнейшей проработки.

**Тема: Одномерные массивы. 1 курс.**

**Определение**: *массив* – это упорядоченный набор фиксированного количества некоторых значений.

**Назначение**: объединить набор близких по назначению объектов в единый объект.

**Виды**: массивы бывают *статические* и *динамические*. Разница между ними в том, что размер статического массива известен на этапе компиляции, поэтому память под него выделяется сразу, что оптимальнее; динамический массив создаётся уже во время исполнения. Во многих прикладных языках (C#/Java) статические массивы вообще отсутствуют.

**Способы задания**: В языке C# (массивы абстрактного типа T):

var mas=new T[]{T1,T2,T3};//объявление вместе с инициализацией

T[] mas=new T[]{T1,T2,T3};//объявление вместе с инициализацией

T[] mas; mas=new T[]{T1,T2,T3};//объявление и инициализация порознь

T[] mas; for(int k=0;i<mas.Lenght;k++) mas[k]=...;//заполнение

В языке D:

int[] array = new int[ 20 ] ; / / Создать массив для 20 целых чисел

auto array = new int[ 20 ] ; / / Создать массив для 20 целых чисел

auto somePrimes = [ 2, 3, 5, 7, 11, 13, 17 ];

Операции с массивами (примеры на D):

1. Получение длины.

auto array = new short[55];

assert( array .length == 55);

1. Срезы:

auto array = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];

/ / Напечатать только вторую половину

writeln (array [S / 2 $]);

1. Копирование:

int [ ] array = [0, 1, 2];

int [ ] subarray = array[1 . . $];

assert ( subarray.length == 2);

subarray[1] = 33;

assert (array[ 2 ] == 33); / / Изменение массива subarray

/ / отразилось на массиве array

1. Проверка на равенство:

auto а = [ "h e llo '', "world"];

auto b = а;

assert ( a i s b); / / Тест пройден, у а и b одни те же границы

assert ( a == b); / / Естественно, тест пройден

b = a.dup;

assert ( a == b); / / Тест пройден, а и b равны,

/ / хотя занимают разные области памяти

assert ( a ! is b); / / Тест пройден, а и b различны, хотя

/ / имеют одинаковое содержимое

1. Конкатенация:

int [ ] а = [0, 10, 20];

int [ ] b = а ^ 42;

assert ( b == [0, 10, 20, 42]);

а = b ^ а ^ 15;

assert ( a . l e n g t h == 8);

1. Поэлементные операции:

auto а = [ 0.5, -0, 5, 1. 5, 2 ];

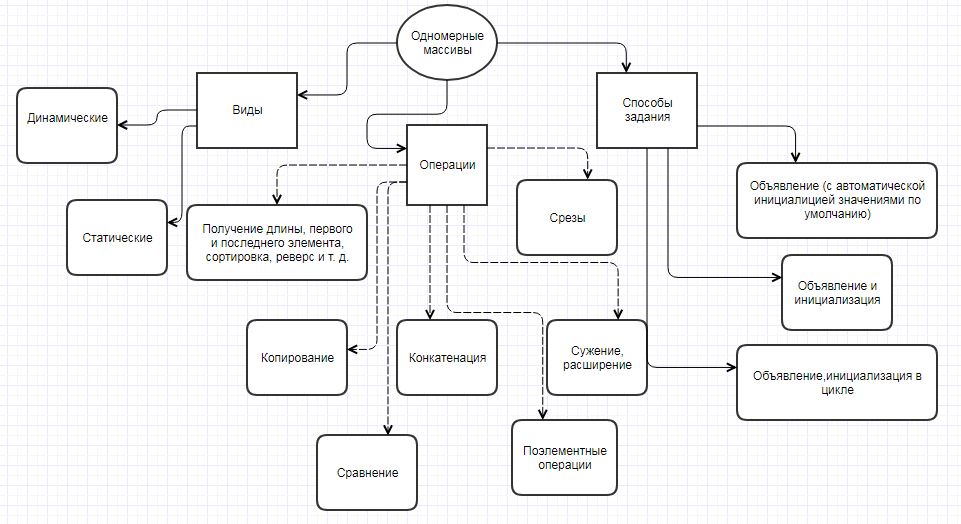
auto b = [ 3.5, 5.5, 4.5, -1 ];

auto с = new double[4]; / / Память под массив должна быть уже выделена

c [] = ( a [ ] + b[]) / 2; / / Рассчитать среднее арифметическое а и b

assert ( c == [ 2 .0 , 2 .5 , 3 .0 , 0 . 5 ]);

**Блок-схема**



**Выводы**. Опорные листы способствуют формированию таких навыков, как анализировать полученные знания, выделять главное и сжимать текст. Также при формировании опорных листов обучающийся лучше усваивает материал, а, сформировав лист, получает уникальный источник информации, к которому будет обращаться в первую очередь.