

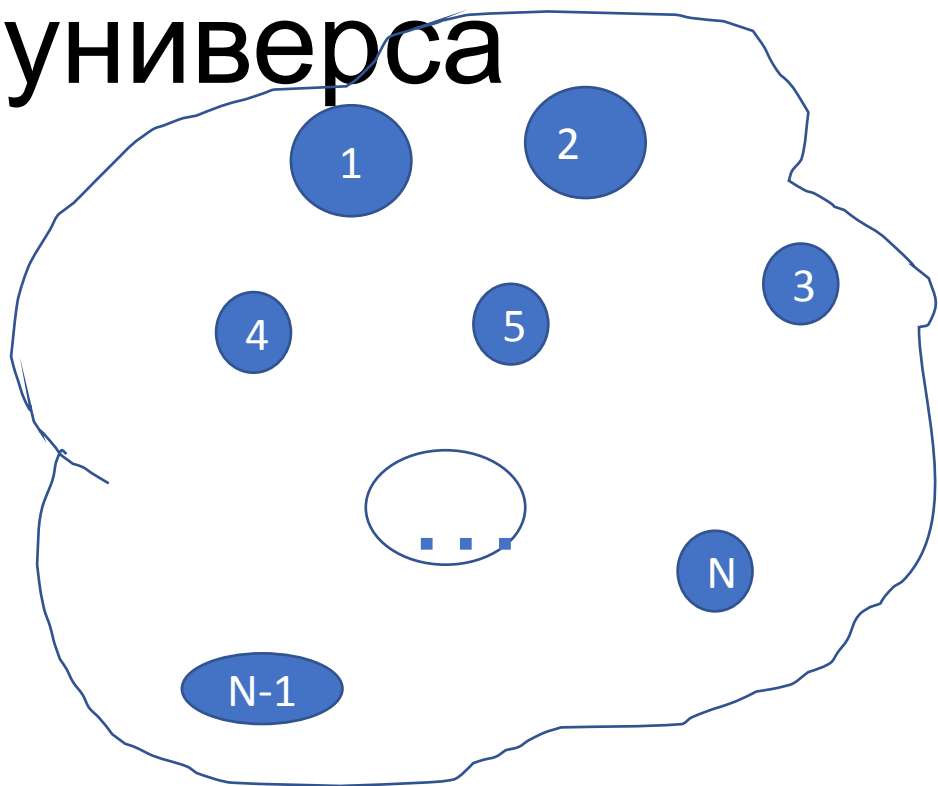
Система операций над множествами

Выбор оптимальной структуры хранения счетных конечных
множеств

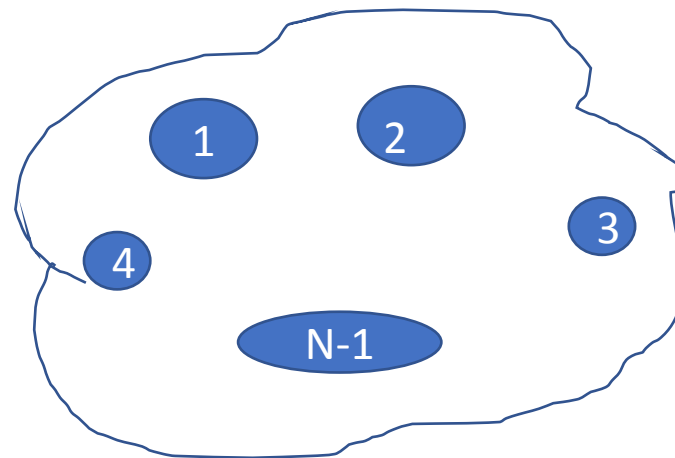
Барышева ИВ

Примеры конечных множеств

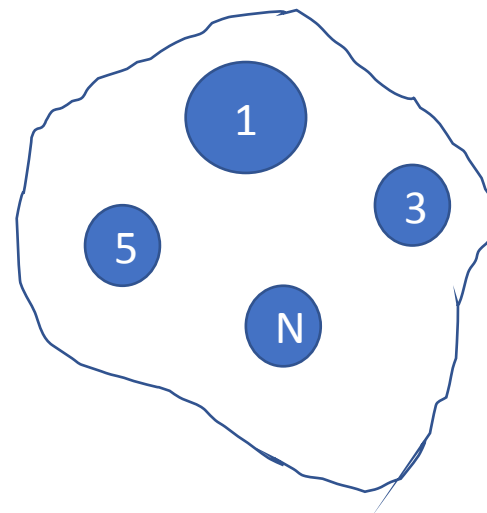
N – размер
универса



универс



Подмножество
А



Подмножество
В

**Размер универса N =
100**

Подмножество A:

1, 5, 20, 7, 11, 75, 41, 50, 80

Подмножество B:

20, 11, 45, 13, 60, 90, 7, 10

Размер универса $N =$

Подмножество $A:$

1, 5, 20, 7, 11, 75, 41, 50, 80

Подмножество $B:$

20, 11, 45, 13, 60, 90, 7, 10

Подмножество $A \cup B:$

1, 5, 7, 10, 11, 13, 20, 41, 45, 50, 60, 75, 80, 90

**Размер универса N =
100**

Подмножество A:

1, 5, 20, 7, 11, 75, 41, 50, 80

Подмножество B:

20, 11, 45, 13, 60, 90, 7, 10

Подмножество $A \cap B$: 7, 11, 20

Размер универса $N =$

100

Подмножество A:

1, 5, 20, 7, 11, 75, 41, 50, 80

Подмножество B:

20, 11, 45, 13, 60, 90, 7, 10

Подмножество $\sim A$:

2, 3, 6, 8, 9, 10, 12, ..., 19, 21, ..., 40, 42, ..., 49,
51, ..., 74, ..., 79, 81, ..., 100

Постановка задачи

1. Разработать структуру хранения множеств
2. Организовать выполнение операций над подмножествами одного универса в соответствии с выбранной структурой хранения
 1. Объединение множеств
 2. Пересечение множеств
 3. Определение дополнения
3. Найти оценки сложности по времени и памяти



Вариант 1

Структура хранения –

массив номеров элементов, входящих в подмножество

тип массива целый

размер массива - ??? 

сложность по памяти - ??  IT

Сложность по времени - квадратичная

Вариант 1

Структура хранения – массив номеров элементов

Сложность по времени

1. Операция объединения:

Действие	СЛОЖНОСТЬ
Все элементы второго подмножества записывается в результат	$O(N/2)$
каждый элемент первого подмножества	$O(N/2)$
ищется во втором подмножестве	$O(N/4)$
если не найден, добавляется в результат	1
ИТОГО	$O(N+N*N/8+1)$

Вариант 1

Структура хранения – массив номеров элементов

Сложность по времени

1. Операция пересечения:

Действие	сложность
каждый элемент первого подмножества	$O(N/2)$
ищется во втором подмножестве	$O(N/4)$
если найден, добавляется в результат	1
ИТОГО	$O(N*N/8+1)$

Вариант 1

Структура хранения – массив номеров элементов

Сложность по времени

1. Операция дополнения:

Действие	СЛОЖНОСТЬ
каждый элемент универса	$O(N)$
ищется в подмножестве	$O(N/2)$
если найден, добавляется в результат	1
ИТОГО	$O(N*N/2+1)$

Технология разработки и тестирования класса

3. Класс подключается к главной программе

```
#include ....  
.....  
Int main( ...)  
{  
  
  
}
```



```
Class Myclass  
{  
  
  
}
```

Технология разработки и тестирования класса

3. В классе прописываются поля (свойства класса)

4. Обязательные методы

```
#include ....  
.....  
Int main( ...)  
{  
  
  
  
  
  
  
  
  
  
}
```



```
Class Myclass  
{  
  
  
  
  
  
  
  
  
  
}
```

Технология разработки и тестирования класса

3. В классе прописываются поля (свойства класса)

4. Обязательные методы:

- Конструктор(ы)

- Деструктор

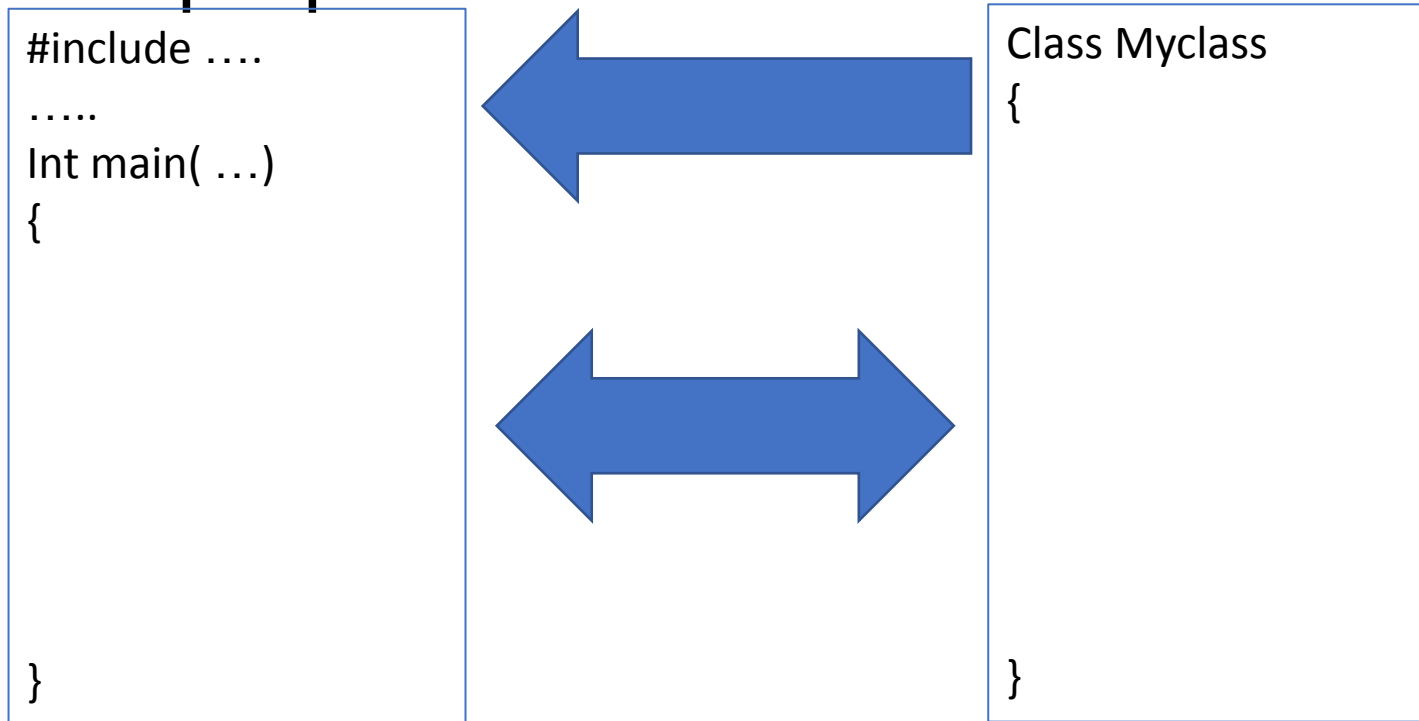
- Конструктор копирования

- Перегрузка оператора присваивания

5. В главной программе объявляются объекты нового класса для тестирования написанных методов

Технология разработки и тестирования класса

6. Параллельно каждый добавленный в класс метод должен быть протестирован, для этого вызывается в главной программе



Задание 1.

Написать и протестировать класс Set, обеспечивающий работу с множествами:

- Ввод множества

- Объединение 2х множеств

- Пересечение 2х множеств

- Дополнение к первому множеству

- Вывод множества

Структура хранения – массив номеров элементов

Вариант 2

Структура хранения – битовая строка, отображенная на память компа

Пример. Размер универса – 36

Подмножество A: 6, 9, 16, 19, 20, 35

Номер элемента универса

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Признак принадлежности элемента к подмножеству,
1-принадлежит, 0-отсутствует

Вариант 2

Структура хранения – битовая строка, отображенная на память компа

Пример. Размер универса – 36

Подмножество В: 2, 7, 12, 13, 22, 23, 24, 35

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0

Вариант 2

Структура хранения – битовая строка,
отображенная на память компа

Пример. Размер универса – 36

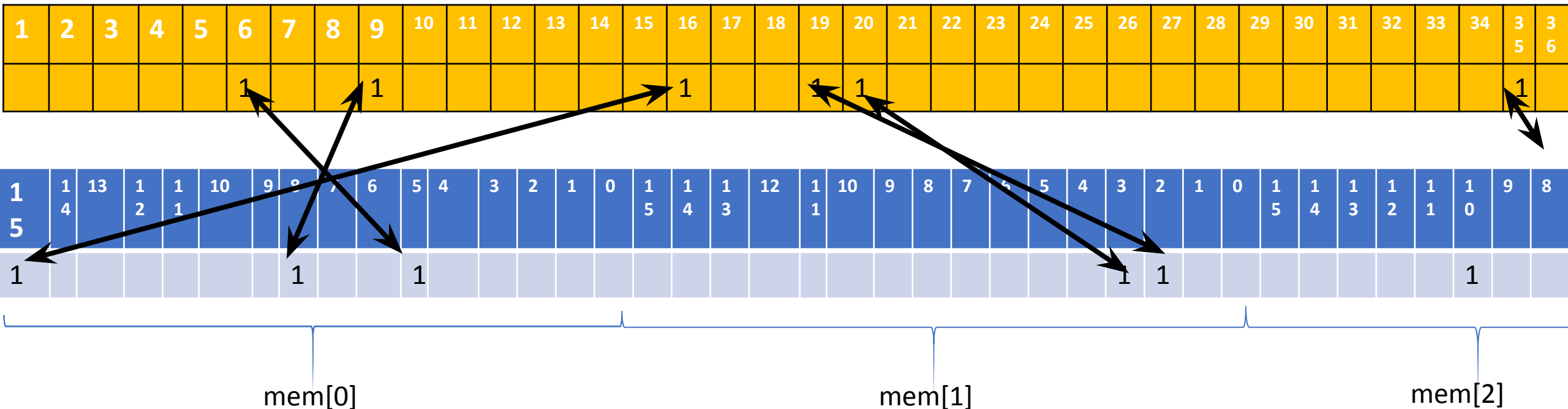


Отображение битовой строки
подмножества A на массив mem типа **char**

Вариант 2

Структура хранения – битовая строка, отображенная на память компа

Пример. Размер универса – 36

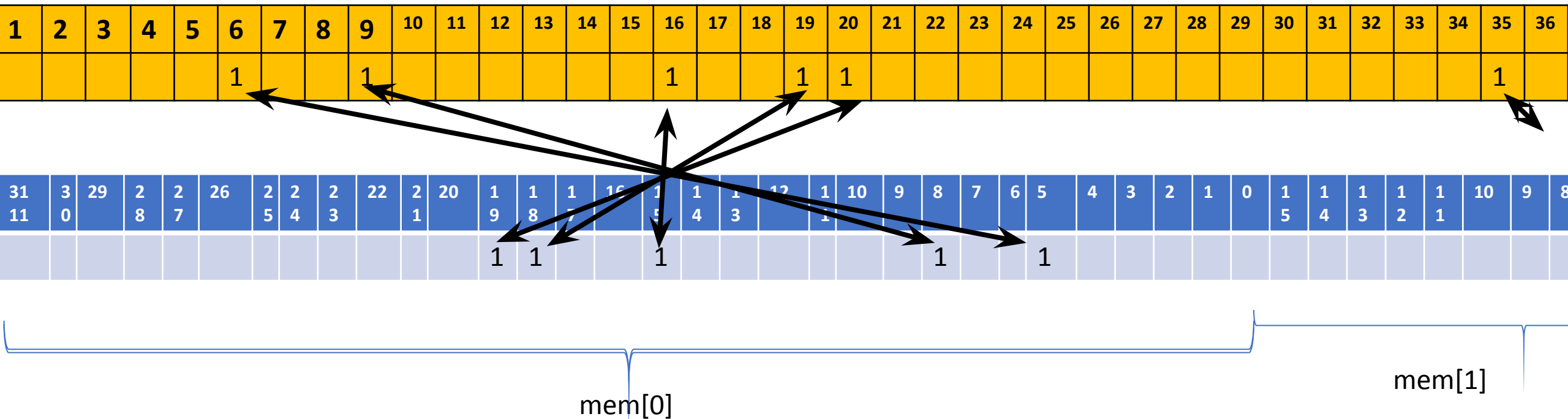


Отображение битовой строки подмножества
A на массив mem типа int16

Вариант 2

Структура хранения – битовая строка, отображенная на память компа

Пример. Размер универса – 36

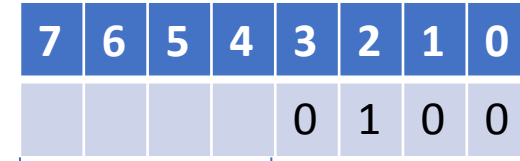


Отображение битовой строки подмножества
A на массив `mem` типа `int32`

Вариант 2

Расположение 35 элемента

массив mem типа char



лишние

mem[4]

массив mem типа int16



лишние

mem[2]



массив mem типа int32

mem[1]

Вариант 2

Размер выделяемой памяти

массив mem типа **char** – size=5, количество байт
5

массив mem типа **int16** - size=3, количество байт
6

массив mem типа **int32** - size=2, количество байт
8

Задание 2.

Исходные данные:

Тип массива

Размер универса

Номер элемента множества

Определить:

Размер выделяемой памяти в байтах

Номер массива mem, содержащий заданный элемент множества

Номер бита, содержащий заданный элемент множества

Оценка сложности по памяти

Размер универсала	сложность	1 вариант	Тип Char	int16	Int32
36	size	36	5	3	2
36	байт	$36*4$	5	6	8
100	size	100	13	7	4
100	байт	$100*4$	13	14	16

Для произвольного N определяем по формуле

Вариант 2

Структура хранения – массив, содержащий битовую

строку
Сложность по времени

1. Операция объединения:

Действие	СЛОЖНОСТЬ
Для каждой пары элементов массивов, содержащих битовые строки соответствующих подмножеств, выполняется битовая операция «ИЛИ»	O(N)
Итого	O(N)

Вариант 2

Структура хранения – массив, содержащий битовую

строку
Сложность по времени

1. Операция пересечения:

Действие	СЛОЖНОСТЬ
Для каждой пары элементов массивов, содержащих битовые строки соответствующих подмножеств, выполняется битовая операция «И»	O(N)
Итого	O(N)

Вариант 2

Структура хранения – массив, содержащий битовую строку

Сложность по времени

1. Операция дополнения:

Действие	СЛОЖНОСТЬ
Для каждого элемента массива, содержащего битовую строку подмножества, выполняется битовая операция «~» - отрицания	O(N)
Итого	O(N)

Пример Размер универса – 36

Подмножество A: 6, 9, 16, 19, 20, 35

Подмножество B: 2, 7, 12, 13, 22, 23, 24, 35

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	1	0	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0

Подмножество A ∪ B: 2, 6, 7, 9, 12, 13, 16, 19, 20, 22, 23, 24, 35

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
0	1	0	0	0	1	1	0	1	0	0	1	1	0	0	1	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	0

Операция “<<” («сдвиг влево») и “>>” («сдвиг вправо»)

Пример 1.

Int16 A=
1<<2



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Результатом операции A=4

Пример 2.


$9_{10} = 1001_2$ после
выполнения операции $\text{int } B = 9 \ll 5$



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0


Операция “(«сдвиг влево» <<“) и “>>” («сдвиг вправо»)

Пример 3. Пусть есть элемент целого массива `mem[i]`, который в двоичном представлении имеет вид



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0

после выполнения операции `int B = mem[i] << 4` будет иметь вид

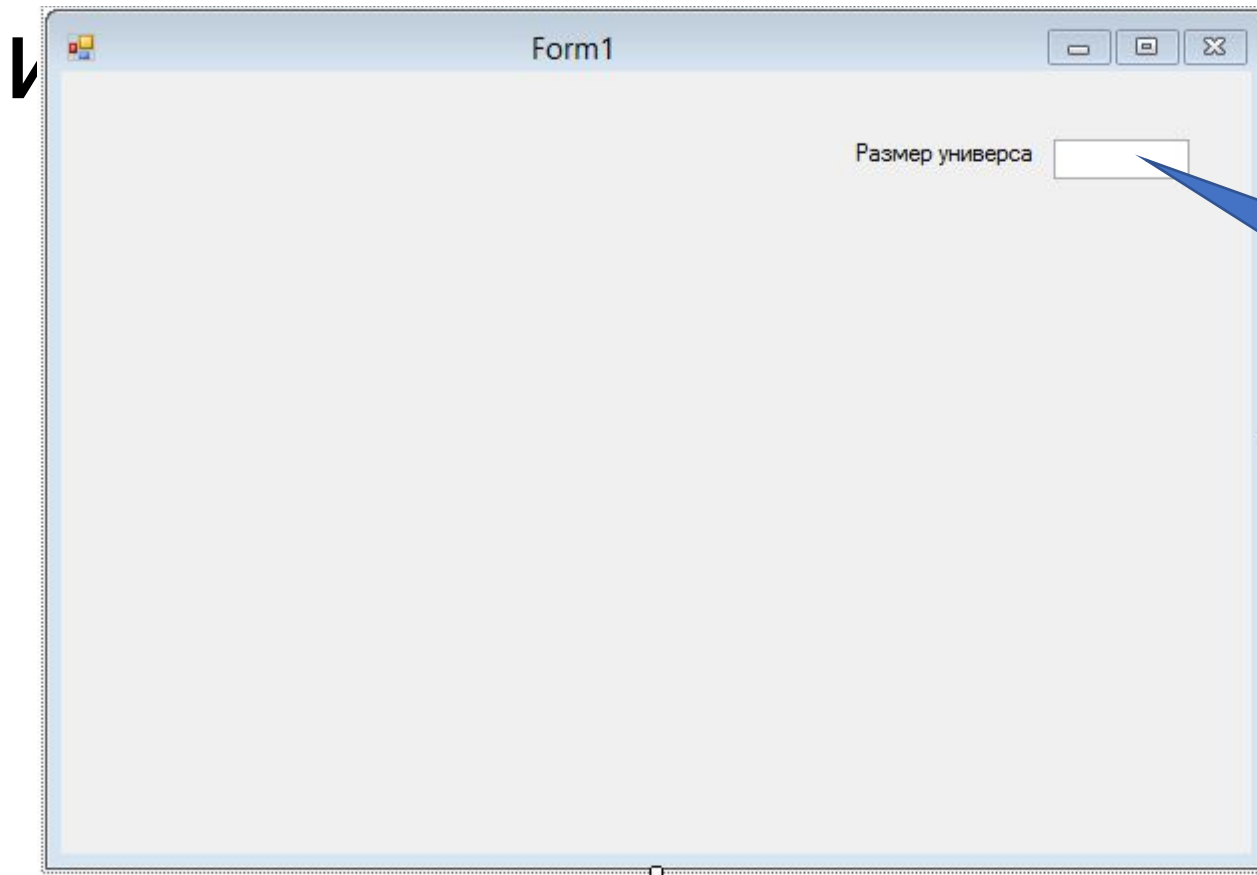


31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0

При этом единица из 29 разряда будет потеряна

Задание на лабораторную работу

1. На WindowsForm разработать



После
установки
размера
универса
открываются
остальные
поля

Поля для ввода подмножеств с контролем и возможностью редактирования

The screenshot shows a Windows application window titled "Form1". It contains the following elements:

- Two rows of input fields labeled "A" and "B". Each row has a long text box, a smaller text box, and two buttons labeled "Add" and "Del".
- A "Размер универса" (Universe Size) label with a text box.
- Three buttons labeled "I", "&", and "~" representing set operations.
- A "Result" label with a text box.

При изменении универса остальные поля зачищаются

Добавка и удаление

операции

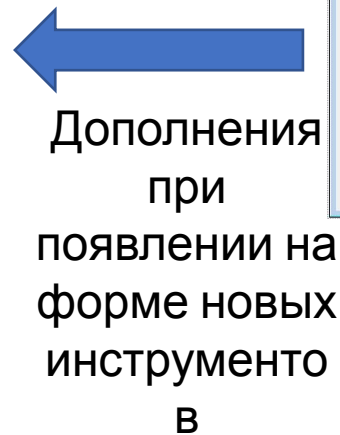
результат

Структура проекта с использованием WindowsForm

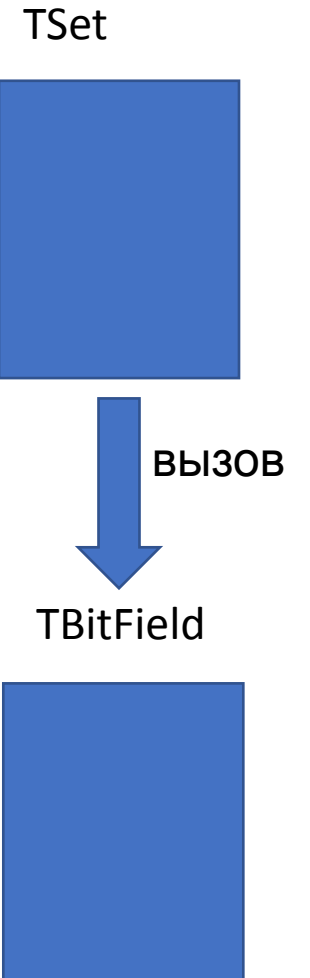
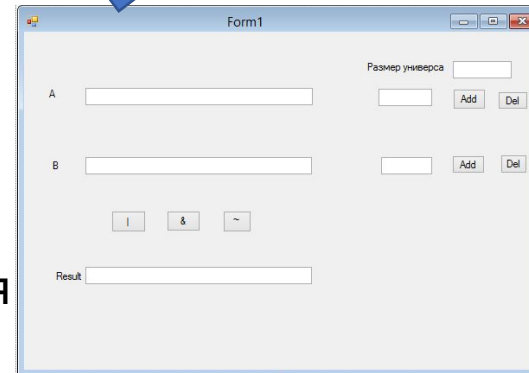
Главная программа формируется автоматически при создании проекта



Form.h формируется и дополняется автоматически

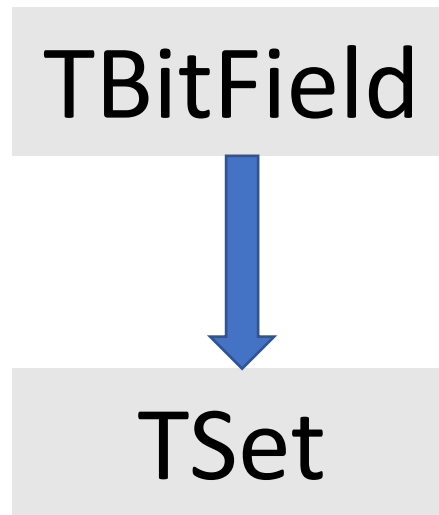


Вызов при наступлении соответствующего события



Задание на лабораторную работу

2. Реализовать структуру классов



Задание на лабораторную работу

3. В классе TBitField должны быть поля

- массив с битовой строкой

- размер массива

Приватные методы по номеру

элемента множества

- вычисление номера бита

- вычисление номера элемента

Задание на лабораторную работу

4. В классе TBitField кроме обязательных должны быть реализованы методы:

- добавить элемент
- удалить элемент
- перегружены операции «И», «ИЛИ», «отрицание»
- преобразование содержимого массива в строку с параметром размера универса

Задание на лабораторную работу

5. В классе TSet должны быть поля
 - объект класса TBitField
 - размер универса

Задание на лабораторную работу

6. В классе TSet кроме обязательных должны быть реализованы методы:

- добавить элемент с контролем
- удалить элемент
- перегружены операции «и», «или», «отрицание»
- ввод подмножества с контролем
- возврат строки с номерами элементов

Порядок сдачи лабораторной работы

1. Класс Set с тестером
2. Контрольная работа
3. Класс TBitField с тестером
4. Класс TSet с тестером
5. Проект на форме