

Лабораторная работа №9

«Перегрузка операций»

Скоробогатов С.Ю.

11 мая 2018 г.

1 Цель работы

Данная работа предназначена для изучения возможностей языка C++, обеспечивающих применение знаков операций к объектам пользовательских типов.

2 Исходные данные

Информация об объявлении и специализации шаблонов дана в лекционном курсе.

3 Задание

Согласно выбранному из таблиц 1–26 описанию требуется составить шаблон класса, перегрузив указанные операции.

Таблица 1: Варианты шаблонов классов

№	Описание	Студент
1	<p>Bool<T> – булевская формула с операциями И, ИЛИ и НЕ, в которой в качестве имён переменных выступают значения типа <i>T</i> (это могут быть, например, строки). Операции, перегружаемые для Bool<T>:</p> <ol style="list-style-type: none"> 1. «*» и «+» – объединение двух формул с помощью операции И и ИЛИ, соответственно; 2. «!» – формирование новой формулы, являющейся отрицанием текущей формулы; 3. «()» – вычисление значения формулы по ассоциативному массиву, отображающему имена переменных в булевские значения. <p>Конструктор класса Bool<T> должен принимать имя переменной и формировать формулу, состоящую из обращения к этой единственной переменной. Подразумевается, что более сложные формулы можно собрать с помощью перегруженных операций.</p>	Бакланов ИУ9-21 11.05
2	<p>Word<Letter> – «слово», представляющее собой последовательность «букв», представленных объектами некоторого класса Letter. Требования к классу Letter: наличие унарного «-» и операции «==» таких, что для любой «буквы» <i>x</i> справедливо равенство $-(-x) == x$. Операции, которые должны быть перегружены для Word<Letter>:</p> <ol style="list-style-type: none"> 1. «+» – конкатенация двух «слов», после которой выполняется «редукция» результирующего «слова», а именно – пары соседних «букв» <i>x</i> и <i>y</i> такие, что $x == -y$, взаимно уничтожаются до тех пор, пока в результирующем «слове» таких пар не останется; 2. унарный «-» – переворачивание слова с одновременной заменой всех «букв» на обратные им «буквы»; 3. «==», «!=». <p>Класс Word<Letter> должен иметь два конструктора: конструктор без параметров создаёт пустое «слово», и конструктор, имеющий параметр Letter, создаёт «слово», состоящее из единственной «буквы».</p>	Боровик ИУ9-21 11.05

Таблица 2: Варианты шаблонов классов

№	Описание	Студент
3	<p>SortedSeq<Symbol> – последовательность отсортированных по возрастанию «символов», представленных объектами некоторого класса Symbol.</p> <p>Требования к классу Symbol: наличие операций «==» и «<», а также операции «~», некоторым образом вычисляющей так называемый «обратный символ». Для любого «символа» x должно быть справедливо, что $\sim(\sim x) == x$.</p> <p>Операции, которые должны быть перегружены для SortedSeq<Symbol>:</p> <ol style="list-style-type: none"> 1. «+=» – добавление «символов» другой последовательности, сопровождаемое «редукцией», представляющей собой удаление из последовательности всех пар взаимнообратных «символов»; 2. «~» – замена всех «символов» в последовательности на обратные им «символы»; 3. «<», «<=», «>» и «>=» – лексикографическое сравнение последовательностей; 4. «==», «!=». <p>Класс SortedSeq<Symbol> должен иметь два конструктора: конструктор без параметров создаёт пустую последовательность, и конструктор, имеющий параметр Symbol, создаёт последовательность, состоящую из единственного «символа».</p>	Богданова ИУ9И-21 11.05
4	<p>PtrStack<T> – стек указателей на структуры типа T.</p> <p>Операции, которые должны быть перегружены для PtrStack<T>:</p> <ol style="list-style-type: none"> 1. «<<» – добавление указателя на вершину стека (push); 2. «>>» – снятие указателя с вершины стека (pop); 3. empty – проверка на пустоту стека; 4. унарный «*» – возвращает значение, адрес которого лежит на вершине стека; 5. «->» – осуществляет доступ к полям структуры, адрес которой лежит на вершине стека. 	Громков ИУ9-21 11.05

Таблица 3: Варианты шаблонов классов

№	Описание	Студент
5	<p>Change<T> – идеальный «размен» суммы денег минимальным количеством рублёвых купюр и монет, в котором количества купюр и монет выражаются целочисленным типом T. Операции:</p> <ol style="list-style-type: none"> 1. «+» – объединение двух разменов; 2. «-» – пересечение двух разменов; 3. «()» – получение количества купюр или монет заданного номинала. 	Дмитриев ИУ9-21 11.05
6	<p>Element<T> – элемент леса непересекающихся множеств с полезной нагрузкой типа T. Операции, которые должны быть перегружены для Element<T>:</p> <ol style="list-style-type: none"> 1. унарный «*» – получение значения, лежащего внутри элемента; 2. «==», «!=» – определение, принадлежат ли два объекта Element<T> одному дереву; 3. унарный «!» – возвращает корень дерева; 4. «<<» – объединение деревьев, которым принадлежат два объекта Element<T>. 	Егорычев ИУ9-21 11.05
7	<p>FibNum<T> – целое число типа T, представленное последовательностью нулей и единиц в фибоначчевой системе счисления. Операции, которые должны быть перегружены для FibNum<T>:</p> <ol style="list-style-type: none"> 1. префиксный и постфиксный «++» – прибавление единицы; 2. «&» – возвращает наибольшее число, составленное из общих для двух чисел фибоначчевых слагаемых; 3. «==», «!=», «<», «<=», «>», «>=»; 4. «T()» – преобразование к типу T. 	Кочанова ИУ9-21 11.05

Таблица 4: Варианты шаблонов классов

№	Описание	Студент
8	<p>$\text{Parcel}\langle T, N \rangle$ – последовательность терминальных и нетерминальных символов, которая получается в процессе вывода предложения некоторого формального языка в соответствии с правилами контекстно-свободной грамматики этого языка. Терминальные символы обозначаются значениями типа T, а нетерминальные – значениями типа N. Подразумевается, что типы T и N различаются. Операции, которые должны быть перегружены для $\text{Parcel}\langle T, N \rangle$:</p> <ol style="list-style-type: none"> 1. «+» – выполняет конкатенацию двух последовательностей с порождением новой последовательности; 2. «+=» – имеет три перегруженные версии: <ol style="list-style-type: none"> (a) добавляет другую последовательность в конец текущей; (b) добавляет терминальный символ в конец текущей последовательности; (c) добавляет нетерминальный символ в конец текущей последовательности; 3. «()» – имея два параметра – нетерминал x и последовательность p, порождает на основе текущей последовательности новую последовательность, в которой самое левое вхождение x заменено на p. <p>Конструктор $\text{Parcel}\langle T, N \rangle$ должен порождать пустую последовательность.</p>	Кузнецов ИУ9-21 11.05
9	<p>$\text{SparseMatrix}\langle T, M, N \rangle$ – разреженная матрица размера $M \times N$ с элементами числового типа T. Представление матрицы должно быть оптимизировано таким образом, чтобы нулевые элементы по возможности не хранились.</p> <p>Операции, которые должны быть перегружены для $\text{SparseArray}\langle T \rangle$:</p> <ol style="list-style-type: none"> 1. «()» – получение ссылки на элемент матрицы (принимает в качестве параметров координаты элемента); 2. «+», «*» – сложение и умножение матриц, умножение на значение типа T; 3. «==», «!=». 	Лобачев ИУ9-21 11.05

Таблица 5: Варианты шаблонов классов

№	Описание	Студент
10	<p>Palindrome<T> – палиндром, составленный из значений типа T. Операции:</p> <ol style="list-style-type: none"> 1. «−» – удаление из палиндрома всех вхождений указанного элемента (например: $abcdcba - b = acdca$); 2. «+» – добавление к палиндрому нового элемента (новый элемент добавляется слева и справа от середины палиндрома: $aa + b = abba$, $aca + b = abcba$); 3. «!» – удаление из палиндрома элементов, расположенных непосредственно слева и справа от середины (например, $!abcba = aca$); 4. «[]» – получение i-го элемента палиндрома. <p>В классе должно быть два конструктора: один создаёт пустой палиндром, а другой – палиндром длины 1.</p>	Маркова ИУ9-21 11.05
11	<p>Seq<T> – последовательность отсортированных по возрастанию значений типа T. (Подразумевается, что для типа T определены операции «<» и «==».) Операции:</p> <ol style="list-style-type: none"> 1. «+» – слияние двух последовательностей в одну; 2. «*» – пересечение двух последовательностей (в результирующей последовательности остаются только элементы, общие для двух пересекаемых последовательностей); 3. «−» – разность последовательностей (результатирующая последовательность содержит элементы, присутствующие в первом операнде и отсутствующие во втором); 4. «[]» – получение i-го элемента последовательности. 	Петрова ИУ9-21 11.05
12	<p>Curve<T> – кривая на плоскости, заданная функцией $y = f(x)$, где x и y – числа с плавающей точкой типа T. Конструктор кривой принимает булевское значение и, в зависимости от этого значения, порождает либо кривую $y = \sin x$, либо кривую $y = \cos x$. Операции:</p> <ol style="list-style-type: none"> 1. «+» – сумма двух кривых: $y = y_1 + y_2 = f_1(x) + f_2(x)$; 2. «−» – разность двух кривых: $y = y_1 - y_2 = f_1(x) - f_2(x)$; 3. «*» – умножение кривой на число: $y = ky_1 = kf_1(x)$; 4. «−» – изменение знака (унарный минус): $y = -y_1 = -f_1(x)$; 5. «!» – дифференцирование: $y = \frac{dy_1}{dx}$; 6. «()» – вычисление y для указанного x. 	Пинская ИУ9-21 11.05

Таблица 6: Варианты шаблонов классов

№	Описание	Студент
13	<p>Set<Letter> – множество «букв», представленных объектами некоторого класса Letter. (Требования к классу Letter: наличие операции «==», а также операции «!», некоторым образом вычисляющей так называемую «обратную букву». Для любой «буквы» x должно быть справедливо, что $!(!x) == x$.) Операции:</p> <ol style="list-style-type: none"> 1. «&=» – добавление «букв» другого множества, сопровождаемое «редукцией», представляющей собой удаление из множества всех пар взаимнообратных «букв»; 2. «&=» – добавление в множество отдельной «буквы», также сопровождаемое «редукцией»; 3. «!» – замена всех «букв» в множестве на обратные им «буквы»; 4. «==», «!=». <p>Класс Set<Letter> должен иметь конструктор без параметров, который создаёт пустое множество.</p>	<p>Поленов ИУ9-21 11.05</p>
14	<p>Series<T> – положительный числовой ряд $\sum_{i=i_0}^{\infty} a_i$, где $a_i \geq 0$ – числа типа T, и $i_0 \geq 0$. Конструктор ряда принимает в качестве параметра i_0 и указатель на функцию, вычисляющую i-тый член ряда. Операции:</p> <ol style="list-style-type: none"> 1. «*» – умножение на число; 2. «+» – сумма двух рядов; 3. «()» – получение i-го члена ряда (если $i < i_0$, операция возвращает 0). 	<p>Родионов ИУ9-21 11.05</p>
15	<p>Shape<T> – множество точек в пространстве $T \times T$, задающих некоторую геометрическую фигуру. Операции, перегружаемые для Shape<T>:</p> <ol style="list-style-type: none"> 1. «+» и «-» – объединение и разность двух множеств; 2. «()» – проверка принадлежности точки множеству. <p>У класса Shape<T> должно быть два конструктора:</p> <ol style="list-style-type: none"> 1. первый конструктор принимает координаты нижней левой и верхней правой вершин прямоугольника, каждая сторона которого параллельна одной из осей координат, и порождает множество точек, принадлежащих этому прямоугольнику; 2. аналогично, второй конструктор порождает множество точек круга по координатам центра и радиусу. 	<p>Санталов ИУ9-21 11.05</p>

Таблица 7: Варианты шаблонов классов

№	Описание	Студент
16	<p>Matrix<T, N> – антидиагональная матрица размера $N \times N$ с элементами типа T. (Все элементы антидиагональной матрицы, кроме лежащих на диагонали, идущей от нижнего левого угла до верхнего правого угла, равны нулю. Матрица должна быть представлена только числами, лежащими на диагонали.) Операции:</p> <ol style="list-style-type: none"> 1. «+» – сумма двух матриц; 2. «*» – произведение двух матриц; 3. «[]» – получение значения элемента, расположенного на i-той строке в j-том столбце. 	Свечникова ИУ9-21 11.05
17	<p>Divs<T, N> – последовательность степеней простых делителей, на которые раскладывается некоторое натуральное число типа T, не превышающее N. Операции, которые должны быть перегружены для Divs<T, N>:</p> <ol style="list-style-type: none"> 1. «*=» – домножение на число, представленное другой последовательностью; 2. «*» – умножение на число, представленное другой последовательностью; 3. «&» – наибольший общий делитель числа, представленного текущей последовательностью, и числа, представленного другой последовательностью; 4. «==», «!=», «<», «<=», «>», «>=»; 5. «T()» – преобразование к типу T. 	Тарасова ИУ9-21 11.05
18	<p>Curve<T> – кривая на плоскости, заданная функцией $y = f(x)$, где x и y – числа с плавающей точкой типа T. Конструктор кривой порождает кривую $y = \exp(x)$. Операции:</p> <ol style="list-style-type: none"> 1. «+» – сумма двух кривых: $y = y_1 + y_2 = f_1(x) + f_2(x)$; 2. «-» – разность двух кривых: $y = y_1 - y_2 = f_1(x) - f_2(x)$; 3. «*» – умножение кривой на число: $y = ky_1 = kf_1(x)$; 4. «-» – изменение знака (унарный минус): $y = -y_1 = -f_1(x)$; 5. «!>» – дифференцирование: $y = \frac{dy_1}{dx}$; 6. «()» – вычисление y для указанного x. 	Узунов ИУ9И-21 11.05

Таблица 8: Варианты шаблонов классов

№	Описание	Студент
19	<p>MergingMap<K,V> – ассоциативный массив, отображающий ключи типа K в значения типа V.</p> <p>Требование к классу V:</p> <ol style="list-style-type: none"> 1. наличие конструктора, принимающего в качестве параметра целое число и в случае, если это число равно 0, порождающего некоторое значение, играющее для типа V роль «нуля»; 2. наличие бинарной операции «+», позволяющей каким-то образом получать «сумму» двух значений (подразумевается, что эта операция является ассоциативной, и вышеупомянутый «ноль» является относительно неё нейтральным элементом). <p>Отметим, что примитивные числовые типы языка C++ удовлетворяют требованиям к классу V и могут быть использованы для проверки работоспособности класса MergingMap<K,V>.</p> <p>Операции, которые должны быть перегружены для MergingMap<K,V>:</p> <ol style="list-style-type: none"> 1. «[]» – возвращает ссылку на значение, связанное с указанным ключом (в случае отсутствия в ассоциативном массиве словарной пары с указанным ключом такая пара автоматически добавляется в массив, причём её значением становится «ноль»); 2. «+» – объединение двух ассоциативных массивов A и B, результатом которого является ассоциативный массив, содержащий такие словарные пары $\langle k, v \rangle$, что k является ключом хотя бы в одном из объединяемых массивов, а $v = A[k] + B[k]$. 3. «==», «!=». <p>Конструктор класса MergingMap<K,V> должен принимать в качестве параметра целое число и создать пустой ассоциативный массив. Параметр конструктора может либо игнорироваться, либо восприниматься как прогнозируемый размер ассоциативного массива для более эффективного выделения памяти.</p> <p>Работоспособность шаблона MergingMap следует проверить для случаев MergingMap<string, int> и MergingMap<string, MergingMap<string, int>>.</p>	<p>Филоненко ИУ9-21 11.05</p>

Таблица 9: Варианты шаблонов классов

№	Описание	Студент
20	<p>NumSet<T> – множество чисел, имеющих тип T. Конструктор класса NumSet<T> принимает в качестве параметров границы интервала $[a, b]$ и формирует множество, содержащее числа, принадлежащие этому интервалу. Подразумевается, что более сложные множества могут быть сконструированы с помощью перегруженных операций. Операции, перегружаемые для NumSet<T>:</p> <ol style="list-style-type: none"> 1. «*» и «+» – пересечение и объединение двух множеств, соответственно; 2. «()» – проверка принадлежности значения типа T множеству. 	Шатнюк ИУ9-21 11.05
21	<p>Program<Statement, Env> – последовательность «команд», представленных объектами некоторого класса Statement, которые можно выполнять в окружении, заданном некоторым классом Env. Подразумевается, что окружение содержит данные, необходимые для работы «команд». Требование к классу Statement: наличие операции «()», которая принимает в качестве параметра ссылку на объект класса Env, выполняет «команду» и возвращает номер команды, на которую должно быть передано управление, или -1, если данная команда завершает закодированную последовательностью программу. Операции, которые должны быть перегружены для Program<Statement, Env>:</p> <ol style="list-style-type: none"> 1. «<<» и «>>» – добавление новой команды в конец или в начало последовательности, соответственно (эти операции возвращают ссылку на текущую последовательность); 2. «()» – выполнение последовательности команд до тех пор, пока некоторая команда не возвратит -1 (принимает в качестве параметра ссылку на объект класса Env); 3. «+» – конкатенация двух последовательностей; 4. «==», «!=». <p>Класс Program<Statement, Env> должен иметь конструктор без параметров, который создаёт пустую последовательность.</p>	Шельдяев ИУ9-21 11.05
22	ЕСЛИ ЕСТЬ ЖЕЛАНИЕ ПОЛУЧИТЬ ЗАДАНИЕ, ПРОСЬБА СВЯЗАТЬСЯ СО МНОЙ, НАПРИМЕР, В КОНТАКТЕ	Ярахмедов ИУ9-21 11.05

Таблица 10: Варианты шаблонов классов

№	Описание	Студент
23	<p>$\text{IntSet}\langle T \rangle$ – множество целых чисел типа T, конструируемое на основе предиката, определяющего принадлежность числа множеству. Конструктор множества принимает в качестве параметра указатель на функцию, принимающую число и возвращающую true или false в зависимости от того, принадлежит число множеству или не принадлежит. Операции:</p> <ol style="list-style-type: none"> 1. «+» – объединение двух множеств; 2. «*» – пересечение двух множеств; 3. «!» – дополнение множества; 4. «*» – умножение всех элементов множества на целое число; 5. «/» – целочисленное деление всех элементов множества на целое число; 6. «()» – проверка принадлежности числа множеству. 	Апахов ИУ9-22 11.05
24	<p>$\text{LinearProgram}\langle \text{Statement}, \text{Env} \rangle$ – последовательность «команд», представленных объектами некоторого класса Statement, которые можно выполнять в окружении, заданном некоторым классом Env. Предполагается, что окружение содержит данные, необходимые для работы «команд».</p> <p>Требование к классу Statement: наличие операции «()», которая принимает в качестве параметра ссылку на объект класса Env, выполняет «команду» и возвращает булевское значение, сообщающее об успешности выполнения команды.</p> <p>Операции, которые должны быть перегружены для $\text{LinearProgram}\langle \text{Statement}, \text{Env} \rangle$:</p> <ol style="list-style-type: none"> 1. «+=» – добавление новой команды в конец последовательности; 2. «()» – выполнение последовательности команд до первой команды, выполненной неуспешно, или до конца (принимает в качестве параметра ссылку на объект класса Env, возвращает bool); 3. «*» – конкатенация двух последовательностей; 4. «==», «!=». <p>Класс $\text{LinearProgram}\langle \text{Statement}, \text{Env} \rangle$ должен иметь конструктор без параметров, который создаёт пустую последовательность.</p>	Бахметьев ИУ9-22 11.05

Таблица 11: Варианты шаблонов классов

№	Описание	Студент
25	<p>LazyMatrix<T> – матрица с элементами типа T неопределённого размера, растущая по мере надобности. Требование к типу T: наличие конструктора по умолчанию. Операции, которые должны быть перегружены для LazyMatrix<T>:</p> <ol style="list-style-type: none"> 1. «()» – получение ссылки элемент с индексами i и j (размер матрицы должен быть автоматически увеличен, если i или j выходят за её пределы); 2. m и n – возвращают количество строк и столбцов, соответственно; 3. «!» – транспонирование матрицы (возвращает новую матрицу). 4. «==», «!=». 	Бойчук ИУ9-22 11.05
26	<p>PtrQueue<T> – очередь указателей на структуры типа T, реализованная через кольцевой буфер. Операции, которые должны быть перегружены для PtrQueue<T>:</p> <ol style="list-style-type: none"> 1. «<<» – добавление указателя на в очередь (enqueue); 2. «!» – вытаскивание указателя из очереди (dequeue); 3. empty – проверка на пустоту очереди; 4. унарный «*» – возвращает значение, адрес которого лежит в начале очереди (туда указывает head); 5. «->» – осуществляет доступ к полям структуры, адрес которой лежит в начале очереди. 	Ботвинников ИУ9-22 11.05
27	<p>LazyArray<T> – массив с элементами типа T неопределённого размера, растущий по мере надобности. Должен быть реализован через класс vector. Требование к типу T: наличие конструктора по умолчанию. Операции, которые должны быть перегружены для LazyArray<T>:</p> <ol style="list-style-type: none"> 1. «[]» – получение ссылки на i-тый элемент массива (размер массива должен быть автоматически увеличен, если i выходит за пределы массива); 2. «()» – формирование подмассива, содержащего элементы с индексами из указанного диапазона (принимает в качестве параметров границы диапазона, возвращает новый LazyArray<T>). 3. «==», «!=». 	Браславский ИУ9-22 11.05

Таблица 12: Варианты шаблонов классов

№	Описание	Студент
28	<p><code>SparseArray<T></code> – разреженный массив, отображающий неотрицательные целые числа в значения типа <code>T</code>. Массив должен быть реализован через хэш-таблицу.</p> <p>Требование к типу <code>T</code>: наличие конструктора по умолчанию (т.к. разреженный массив должен уметь создавать значения типа <code>T</code>).</p> <p>Операции, которые должны быть перегружены для <code>SparseArray<T></code>:</p> <ol style="list-style-type: none"> 1. «<code>[]</code>» – получение ссылки на i-тый элемент массива; 2. «<code>()</code>» – формирование подмассива, содержащего элементы с индексами из указанного диапазона (принимает в качестве параметров границы диапазона, возвращает новый <code>SparseArray<T></code>). 3. «<code>==</code>», «<code>!=</code>». 	Гавриленко ИУ9-22 11.05
29	<p><code>Function<A, R, F></code> – «обёртка» вокруг функции (или объекта класса с перегруженной операцией «<code>()</code>») типа <code>F</code>, принимающей параметр типа <code>A</code> и возвращающей значение типа <code>R</code>.</p> <p>Операции, перегружаемые для <code>Function<A, R, F></code>:</p> <ol style="list-style-type: none"> 1. «<code>()</code>» – вызов функции (принимает значение типа <code>A</code> и возвращает значение типа <code>R</code>); 2. «<code>*</code>» – композиция двух функций. <p>Конструктор класса <code>Function<A, R, F></code> должен принимать параметр типа <code>F</code>.</p> <p><i>Указание:</i> для решения задачи можно составить шаблон класса <code>AbstractFunction<A, R></code> и сделать класс <code>Function<A, R, F></code> наследником <code>AbstractFunction<A, R></code>. Тогда композицию функций $f : A \rightarrow B$ и $g : B \rightarrow R$ можно будет представить объектом класса <code>Composition<A, B, R></code>, также являющегося наследником <code>AbstractFunction<A, R></code>.</p>	Гавриловский ИУ9-22 11.05
30	ЕСЛИ ЕСТЬ ЖЕЛАНИЕ ПОЛУЧИТЬ ЗАДАНИЕ, ПРОСЬБА СВЯЗАТЬСЯ СО МНОЙ, НАПРИМЕР, В КОНТАКТЕ	Гулин ИУ9-22 11.05

Таблица 13: Варианты шаблонов классов

№	Описание	Студент
31	<p>Ring<T> – кольцевой двунаправленный список с ограничителем, в узлах которого хранятся значения типа T.</p> <p>Операции, которые должны быть перегружены для Ring<T>:</p> <ol style="list-style-type: none"> 1. «+» – конкатенация двух списков; 2. «<<» и «>>» – добавление нового элемента в конец или начало списка, соответственно; 3. «*» – поиск элемента, содержащего указанное значение (возвращает bool); 4. «/=» – удаление элемента, содержащего указанное значение; 5. «==», «!=». <p>Класс Ring<T> должен иметь конструктор без параметров, который создаёт пустой список.</p>	Даровская ИУ9-22 11.05
32	ЕСЛИ ЕСТЬ ЖЕЛАНИЕ ПОЛУЧИТЬ ЗАДАНИЕ, ПРОСЬБА СВЯЗАТЬСЯ СО МНОЙ, НАПРИМЕР, ВКОНТАКТЕ	Испирян ИУ9-22 11.05
33	<p>Palindrome<T> – палиндром, составленный из значений типа T.</p> <p>Операции:</p> <ol style="list-style-type: none"> 1. «+» – добавление к палиндрому нового элемента (новый элемент добавляется в начало палиндрома и в его конец: $bb + a = abba$); 2. «!» – удаление из палиндрома крайних элементов (например, $!abba = bb$); 3. «/» – удаление из левого операнда элементов, принадлежащих правому операнду (например: $abcdcba/bdb = acca$); 4. «[]» – получение ссылки на i-ый элемент палиндрома (учесть возможность изменения значения элемента через эту ссылку). <p>В классе должно быть два конструктора: один создаёт пустой палиндром, а другой – палиндром, состоящий из единственного элемента.</p>	Климова ИУ9-22 11.05

Таблица 14: Варианты шаблонов классов

№	Описание	Студент
34	<p>$\text{Seq}\langle T \rangle$ – последовательность значений типа T. (Подразумевается, что для типа T определена операция «$=$».) Операции:</p> <ol style="list-style-type: none"> 1. «$+$» – конкатенация двух последовательностей; 2. «$/$» – деление последовательности на k равных по длине частей (в результате получается вектор последовательностей, длины которых различаются не более, чем на 1); 3. «$!$» – удаление из последовательности дублирующихся значений (в результате формируется новая последовательность); 4. «$[]$» – получение i-го элемента последовательности. 	Котова ИУ9-22 11.05
35	<p>$F\langle N, T \rangle$ – арифметическая формула с операциями сложения, вычитания, умножения и деления, в которой в качестве имён переменных выступают значения типа N (это могут быть, например, строки), а в качестве констант – значения числового типа T. Операции, перегружаемые для $F\langle N, T \rangle$:</p> <ol style="list-style-type: none"> 1. «$+$», «$-$», «$*$» и «$/$» – объединение двух формул с помощью соответствующих операций; 2. унарный «$-$» – формирование новой формулы, являющейся отрицанием текущей формулы; 3. «$()$» – формирование новой формулы, которое заключается в выполнении двух действий: <ol style="list-style-type: none"> (a) подстановка в текущую формулу значений переменных, взятых из ассоциативного массива, отображающего имена переменных в значения типа T; (b) упрощение формулы, достигаемое за счёт вычисления значений подвыражений, не содержащих имена переменных; 4. «$<<<$» – вывод формулы в поток вывода. <p>У класса $F\langle N, T \rangle$ должно быть два конструктора: первый принимает имя переменной и формирует формулу, состоящую из обращения к этой единственной переменной; а второй принимает константу и формирует формулу, состоящую из этой константы. Подразумевается, что более сложные формулы можно собрать с помощью перегруженных операций.</p>	Мамаев ИУ9-22 11.05

Таблица 15: Варианты шаблонов классов

№	Описание	Студент
36	<p>Relation<T> – отношение на множестве значений типа T. Операции, которые должны быть перегружены для Relation<T>:</p> <ol style="list-style-type: none"> 1. «+» – объединение двух отношений; 2. «*» – пересечение двух отношений; 3. «~» – транзитивное замыкание отношения; 4. «()» – проверка принадлежности указанной пары значений отношению (имеет два параметра типа T, возвращает bool); 5. «==», «!=». 	Мирзоева ИУ9-22 11.05
37	<p>NNF<T> – булевская формула с операциями И, ИЛИ и НЕ, в которой в качестве имён переменных выступают значения типа T (это могут быть, например, строки). Формула должна находиться в так называемой <i>нормальной форме отрицания</i> (Negation Normal Form), в которой операция НЕ применяется только к значениям переменных. Операции, перегружаемые для NNF<T>:</p> <ol style="list-style-type: none"> 1. «*» и «+» – объединение двух формул с помощью операции И и ИЛИ, соответственно; 2. «!» – формирование новой формулы, являющейся отрицанием текущей формулы (здесь надо не забыть о нормальной форме отрицания!); 3. «()» – вычисление значения формулы по ассоциативному массиву, отображающему имена переменных в булевские значения; 4. «<<» – вывод формулы в поток вывода. <p>Конструктор класса NNF<T> должен принимать имя переменной и формировать формулу, состоящую из обращения к этой единственной переменной. Подразумевается, что более сложные формулы можно собрать с помощью перегруженных операций.</p>	Пичугин ИУ9-22 11.05
38	<p>Curve<T> – кривая на плоскости, заданная функцией $\vec{r} = f(t)$, где \vec{r} – радиус-вектор с координатами типа T, а t – параметр, также задаваемый числом типа T. Конструктор траектории принимает в качестве параметра указатель на функцию f. Операции:</p> <ol style="list-style-type: none"> 1. «+» – сумма двух кривых: $\vec{r} = \vec{r}_1 + \vec{r}_2 = f_1(t) + f_2(t)$; 2. «*» – умножение кривой на число: $\vec{r} = k\vec{r}_1 = kf_1(t)$; 3. «()» – вычисление радиус-вектора для указанного t. 	Прийма ИУ9-22 11.05

Таблица 16: Варианты шаблонов классов

№	Описание	Студент
39	<p>Seq<Element> – последовательность элементов, представленных объектами некоторого класса Element.</p> <p>Требования к классу Element: наличие операции «!», некоторым образом вычисляющей «обратный» элемент, и операции «==» таких, что для любого элемента x справедливо равенство $!(!x) == x$.</p> <p>Операции, которые должны быть перегружены для Seq<Element>:</p> <ol style="list-style-type: none"> 1. «*» – конкатенация двух последовательностей, после которой выполняется «редукция» результирующей последовательности, а именно – пары соседних элементов x и y такие, что $x == !y$, взаимно уничтожаются до тех пор, пока в результирующей последовательности таких пар не останется; 2. «!» – переворачивание последовательности с одновременной заменой всех элементов на обратные им элементы; 3. «<<» и «>>» – добавление элемента в конец и в начало последовательности, соответственно, после которого выполняется «редукция»; 4. «==», «!=». <p>Класс Seq<Element> должен иметь конструктор без параметров, который создаёт пустую последовательность.</p>	Ростецкий ИУ9-22 11.05
40	<p>Seq<T> – последовательность чисел типа T, конструируемая как арифметическая прогрессия:</p> $a(n) = a_0 + dn.$ <p>Конструктор последовательности принимает в качестве параметров a_0 и d. Операции:</p> <ol style="list-style-type: none"> 1. «+» – сумма двух последовательностей, определяемая как $a = a_1 + a_2 \iff \forall n \geq 0 : a(n) = a_1(n) + a_2(n);$ 2. «*» – умножение последовательности на число, определяемое как $a = k \star a_1 \iff \forall n \geq 0 : a(n) = ka_1(n);$ 3. «/» – обнуление всех членов последовательности, начиная с заданного номера, определяемое как $a = a_1/n_0 \iff \forall n \geq 0 : a(n) = \begin{cases} a_1(n), & n < n_0; \\ 0, & n \geq n_0; \end{cases}$ 4. «()» – вычисление i-го члена последовательности. 	Рыбаков ИУ9-22 11.05

Таблица 17: Варианты шаблонов классов

№	Описание	Студент
41	<p>Table<T> – таблица, в ячейках которой размещены значения числового типа T, а столбцы имеют строковые имена. Конструктор класса Table<T> должен формировать пустую таблицу, состоящую из одного столбца с указанным именем. Подразумевается, что более сложные таблицы можно построить из столбцов с помощью перегруженных операций:</p> <ol style="list-style-type: none"> 1. «+» – соединение двух таблиц, в результате которой формируется третья таблица по следующим правилам: <ol style="list-style-type: none"> (a) множество имён столбцов результирующей таблицы является объединением множеств имён столбцов исходных таблиц; (b) если число строк в одной из соединяемых таблиц меньше числа строк во второй, то будем считать, что она содержит нужное количество строк, забитых нулями и расположенных внизу таблицы; (c) если имеется имя столбца, присутствующее как в первой из соединяемых таблиц, так и во второй, то соответствующий столбец результирующей таблицы должен содержать суммы соответствующих столбцов исходных таблиц; (d) столбец, имя которого присутствует только в одной из соединяемых таблиц, просто копируется в результирующую таблицу; 2. «%» – добавление в конец таблицы строки, во всех ячейках которой находится указанное значение; 3. «[]» – получение временного объекта, обозначающего строку таблицы, к которому, в свою очередь, может быть применена операция «[]», принимающая в качестве параметра имя столбца, для получения ссылки на ячейку строки; 4. «size» – получение количества строк в таблице; 5. «<<» – вывод таблицы в поток вывода. 	<p>Спиридонова ИУ9-22 11.05</p>
42	<p>ЕСЛИ ЕСТЬ ЖЕЛАНИЕ ПОЛУЧИТЬ ЗАДАНИЕ, ПРОСЬБА СВЯЗАТЬСЯ СО МНОЙ, НАПРИМЕР, В КОНТАКТЕ</p>	<p>Актюрк ИУ9-23 11.05</p>

Таблица 18: Варианты шаблонов классов

№	Описание	Студент
43	<p>$\text{Matrix}<T, M, N>$ – матрица размера $M \times N$, элементами которой являются числа типа T, заданная функцией $f(i, j)$, возвращающей значение элемента, находящегося на i-той строке в j-том столбце. Конструктор матрицы принимает в качестве параметра указатель на функцию f. Операции:</p> <ol style="list-style-type: none"> 1. «+» – сумма двух матриц; 2. «*» – произведение двух матриц; 3. «[]» – получение значения элемента, расположенного на i-той строке в j-том столбце. <p>Предполагается, что матрица может иметь очень большие размеры. Поэтому заранее вычислять значения элементов матрицы и хранить их в массиве нецелесообразно – массив может не поместиться в оперативной памяти.</p>	Артеменко ИУ9-23 11.05
44	<p>$\text{Polyline}<P>$ – ломаная линия, состоящая из точек типа P. Требование к типу P: наличие метода <code>dist</code>, вычисляющего расстояние до другой точки. Операции, которые должны быть перегружены для $\text{Polyline}<P>$:</p> <ol style="list-style-type: none"> 1. «<<» и «>>» – добавление точки в конец или в начало ломаной, соответственно (операции возвращают ссылку на текущую ломаную); 2. «[]» – возвращает ссылку на i-тую точку ломаной; 3. <code>count</code> – возвращает количество точек ломаной; 4. «==», «!=», «<», «<=», «>», «>=» – ломаные сравниваются по длине. 	Бакланова ИУ9-23 11.05
45	<p>$\text{LinearIneq}<T, N>$ – линейное неравенство вида $a_1x_1 + a_2x_2 + \dots + a_Nx_N \bowtie b$, в котором a_i, x_i и b имеют тип T, а \bowtie – это $<$, \leq, $>$ или \geq. Операции, перегружаемые для $\text{LinearIneq}<T, N>$:</p> <ol style="list-style-type: none"> 1. «*» – умножение левой и правой части на число, отличное от нуля; 2. «!» – формирование отрицания неравенства (т.е., например, из $x \leq 5$ получается $x > 5$); 3. «+» – сложение двух неравенств; 4. «()» – проверка справедливости неравенства для вектора значений переменных. 	Богданов ИУ9-23 11.05

Таблица 19: Варианты шаблонов классов

№	Описание	Студент
46	<p>LinearEq<T> – линейное равенство вида $a_1x_1 + a_2x_2 + \dots + a_nx_n \doteq b$, в котором a_i, x_i и b имеют тип T, а \doteq – это $=$ или \neq.</p> <p>Операции, перегружаемые для LinearEq<T>:</p> <ol style="list-style-type: none"> 1. «*» – умножение левой и правой части на число; 2. «!» – формирование отрицания равенства (т.е., например, из $x = 5$ получается $x \neq 5$); 3. «[]» – получение ссылки на i-тый коэффициент равенства (при $i = 0$ возвращается ссылка на b); 4. «()» – проверка справедливости равенства для вектора значений переменных. 	<p>Бокарев ИУ9-23 11.05</p>
47	<p>Set<T> – множество значений типа T.</p> <p>Операции, перегружаемые для Set<T>:</p> <ol style="list-style-type: none"> 1. «*» и «+» – пересечение и объединение двух множеств, соответственно; 2. «!» – дополнение множества до универсального множества, содержащего все возможные значения типа T; 3. «()» – проверка принадлежности значения типа T множеству. <p>Конструктор класса Set<T> должен принимать значение типа T и формировать множество, состоящее из этого единственного значения. Подразумевается, что более сложные множества можно собрать с помощью перегруженных операций.</p>	<p>Бостаджян ИУ9-23 11.05</p>
48	<p>ЕСЛИ ЕСТЬ ЖЕЛАНИЕ ПОЛУЧИТЬ ЗАДАНИЕ, ПРОСЬБА СВЯЗАТЬСЯ СО МНОЙ, НАПРИМЕР, В КОНТАКТЕ</p>	<p>Дурдымура- дова ИУ9-23 11.05</p>

Таблица 20: Варианты шаблонов классов

№	Описание	Студент
49	<p>$\text{PolyIneq}\langle T, N \rangle$ – неравенство вида $a_1x^N + a_2x^{N-1} + \dots + a_Nx \bowtie b$, в котором a_i, x_i и b имеют тип T, а \bowtie – это $<$, \leq, $>$ или \geq. Операции, перегружаемые для $\text{PolyIneq}\langle T, N \rangle$:</p> <ol style="list-style-type: none"> 1. «*» – умножение левой и правой части на число, отличное от нуля; 2. «!» – формирование отрицания неравенства (т.е., например, из $x^2 + 2x \leq 5$ получается $x^2 + 2x > 5$); 3. «+» – сложение двух неравенств; 4. «()» – проверка справедливости неравенства для указанного значения переменной x. 	Жданов ИУ9-23 11.05
50	<p>$\text{IneqSys}\langle T, N \rangle$ – неизменяемая система неравенств вида $a_1x_1 + a_2x_2 + \dots + a_Nx_N \leq b$, где a_i и b имеют тип T. Операции, перегружаемые для $\text{IneqSys}\langle T, N \rangle$:</p> <ol style="list-style-type: none"> 1. «+» – объединение двух систем в одну; 2. «[]» – принимает число i и возвращает систему, полученную из данной системы путём присвоения нулевого значения i-той переменной; 3. «()» – проверка, удовлетворяет ли вектор значений переменных системе неравенств. <p>Конструктор класса $\text{IneqSys}\langle T, N \rangle$ должен принимать массив коэффициентов a_i и свободный член b и порождать систему, состоящую из одного неравенства.</p>	Жулева ИУ9-23 11.05
51	<p>$\text{Palindrome}\langle T, N \rangle$ – палиндром размера N, составленный из значений типа T. Операции:</p> <ol style="list-style-type: none"> 1. «+» – объединение двух палиндромов (при объединении левый операнд вставляется в середину правого, например: $qwq + abba = abqwqba$; данная операция доступна только в случае, если размер правого операнда – чётный); 2. «!» – выворачивание палиндрома наизнанку (например, $!abcdcba = cbadabc$); 3. «[]» – получение ссылки на i-ый элемент палиндрома (учесть возможность изменения значения элемента через эту ссылку). <p>Конструктор класса должен принимать в качестве параметра одно значение типа T и порождать палиндром, состоящий из N вхождений этого значения.</p>	Зайцев ИУ9-23 11.05

Таблица 21: Варианты шаблонов классов

№	Описание	Студент
52	<p>$\text{Seq}\langle T \rangle$ – последовательность отсортированных по возрастанию значений типа T. (Подразумевается, что для типа T определены операции «$<$» и «$=$».) Операции:</p> <ol style="list-style-type: none"> 1. «$+$» – слияние двух последовательностей в одну; 2. «$*$» – умножение на число k (каждый элемент исходной последовательности повторяется в результирующей последовательности k раз); 3. «$!$» – удаление из последовательности дублирующихся значений (в результате формируется новая последовательность); 4. «$[\]$» – получение i-го элемента последовательности. 	<p>Конюхов ИУ9-23 11.05</p>
53	<p>$\text{Seq}\langle T, N \rangle$ – последовательность длины N, состоящая из отсортированных по возрастанию значений типа T. (Подразумевается, что для типа T определена операция «$<$».) Операции:</p> <ol style="list-style-type: none"> 1. «$+$» – слияние двух последовательностей в одну (правый операнд – последовательность длины N_2, результирующая последовательность имеет длину $N + N_2$); 2. «$*$» – умножение на число k (каждый элемент исходной последовательности повторяется в результирующей последовательности k раз; тем самым, размер результирующей последовательности – kN); 3. «$[\]$» – получение i-го элемента последовательности. 	<p>Курушин ИУ9-23 11.05</p>

Таблица 22: Варианты шаблонов классов

№	Описание	Студент
54	<p>$\text{ApproxNum}\langle T \rangle$ – число вида $a + k\delta$, где a и k имеют тип T, а δ – неизвестное бесконечно малое число.</p> <p>Сложение, умножение на число типа T и сравнение чисел $\text{ApproxNum}\langle T \rangle$ определяется правилами:</p> <ol style="list-style-type: none"> 1. $(a_1 + k_1\delta) + (a_2 + k_2\delta) \equiv a_1 + a_2 + (k_1 + k_2)\delta$; 2. $c(a + k\delta) \equiv ca + ck\delta$; 3. $a_1 + k_1\delta < a_2 + k_2\delta$ тогда и только тогда, когда либо $a_1 < a_2$, либо $(a_1 = a_2) \wedge (k_1 < k_2)$. <p>Операции, перегружаемые для $\text{ApproxNum}\langle T \rangle$:</p> <ol style="list-style-type: none"> 1. «*=» – домножение на число типа T; 2. «+=», «-=» – прибавление и вычитание числа типа $\text{ApproxNum}\langle T \rangle$; 3. «+», «-» – сложение и вычитание двух чисел типа $\text{ApproxNum}\langle T \rangle$; 4. «*» – умножение числа типа $\text{ApproxNum}\langle T \rangle$ на число типа T, а также умножение числа типа T на число типа $\text{ApproxNum}\langle T \rangle$; 5. «==», «!=», «<», «<=», «>», «>=» – сравнение чисел типа $\text{ApproxNum}\langle T \rangle$. 	<p>Лысенко ИУ9-23 11.05</p>
55	<p>$\text{Node}\langle T \rangle$ – элемент однонаправленного списка, в котором хранится значение типа T. Список, как это принято в языке Лисп, должен заканчиваться специальным элементом, который должен быть представлен статическим полем $\text{Node}::\text{NIL}$.</p> <p>Операции, перегружаемые для $\text{Node}\langle T \rangle$:</p> <ol style="list-style-type: none"> 1. «<<» – добавление нового элемента в начало списка (головной элемент списка – левый операнд, значение типа T – правый операнд); 2. «++» – получение хвоста списка (для $\text{Node}::\text{NIL}$ должна порождать исключение); 3. унарная «*» – получение значения, хранящегося в элементе списка (для $\text{Node}::\text{NIL}$ должна порождать исключение); 4. «+=» – добавление другого списка в конец текущего списка. 	<p>Петров ИУ9-23 11.05</p>

Таблица 23: Варианты шаблонов классов

№	Описание	Студент
56	<p>ProgressionSet<T> – множество целых чисел типа T. Конструктор класса ProgressionSet<T> принимает в качестве параметров первый член и разность арифметической прогрессии и формирует множество, содержащее числа, принадлежащие этой прогрессии. Подразумевается, что более сложные множества могут быть сконструированы с помощью перегруженных операций. Операции, перегружаемые для ProgressionSet<T>:</p> <ol style="list-style-type: none"> 1. «*» и «+» – пересечение и объединение двух множеств, соответственно; 2. «()» – проверка принадлежности значения типа T множеству. 	Пинчук ИУ9-23 11.05
57	<p>StackProgram<T> – программа для стековой машины, оперирующей со значениями типа T. Программа представляет собой последовательность из следующих команд, каждая из которых должна быть представлена отдельным классом:</p> <ol style="list-style-type: none"> 1. «LoadConst k» – загрузка на стек константы k типа T; 2. «LoadVar x» – загрузка на стек переменной с именем x; 3. «BinaryOp op» – выполнение операции op (сложение, вычитание, умножение или деление) над двумя верхними элементами стека (эти элементы снимаются со стека, а результат кладётся на стек). <p>У класса StackProgram<T> должно быть два конструктора: первый принимает константу и формирует программу, состоящую из единственной команды «LoadConst k»; а второй принимает имя переменной и формирует программу, состоящую из единственной команды «LoadVar x». Подразумевается, что более сложные программы можно собрать с помощью перегруженных операций:</p> <ol style="list-style-type: none"> 1. «+», «-», «*» и «/» – объединение двух программ с помощью соответствующих операций; 2. унарный «-» – формирование новой программы, вычисляющей такое же значение, что и текущая программ, но с противоположным знаком; 3. «[]» – получение ссылки на i-тую команду; 4. «length» – получение количества команд в программе; 5. «()» – выполнение программы (операция получает ассоциативный массив, отображающий имена переменных в их значения); 6. «<<» – вывод программы в поток вывода. 	Подольный ИУ9-23 11.05

Таблица 24: Варианты шаблонов классов

№	Описание	Студент
58	<p>NumSet<T> – множество целых чисел типа T. Конструктор класса NumSet<T> принимает в качестве параметров некоторое число k и формирует множество, содержащее все числа, кратные числу k. Подразумевается, что более сложные множества могут быть сконструированы с помощью перегруженных операций. Операции, перегружаемые для NumSet<T>:</p> <ol style="list-style-type: none"> 1. «*» и «+» – пересечение и объединение двух множеств, соответственно; 2. «()» – проверка принадлежности значения типа T множеству. 	Рогов ИУ9-23 11.05
59	<p>Curve<T> – кривая на плоскости, заданная функцией $y = f(x)$, где x и y – числа типа T. Конструктор кривой принимает в качестве параметра указатель на функцию f. Операции:</p> <ol style="list-style-type: none"> 1. «+» – сумма двух кривых: $y = y_1 + y_2 = f_1(x) + f_2(x)$; 2. «-» – разность двух кривых: $y = y_1 - y_2 = f_1(x) - f_2(x)$; 3. «*» – умножение кривой на число: $y = ky_1 = kf_1(x)$; 4. «()» – вычисление y для указанного x. 	Снегур ИУ9-23 11.05
60	<p>Word<Symbol> – «слово», представляющее собой последовательность «символов», представленных объектами некоторого класса Symbol. Требование к классу Symbol: наличие операции «==». Операции, которые должны быть перегружены для Word<Symbol>:</p> <ol style="list-style-type: none"> 1. «&» – конкатенация двух «слов», после которой выполняется «редукция» результирующего «слова», а именно – пары соседних «символов» x и y такие, что $x == y$, взаимно уничтожаются до тех пор, пока в результирующем «слове» таких пар не останется; 2. унарный «-» – переворачивание слова; 3. «+» – добавление «символа» в конец или в начало «слова», после которого выполняется «редукция» (если левый операнд – «символ», то он добавляется в начало второго операнда, который должен быть «словом»; если же левый операнд – «слово», то «символ» добавляется в его конец); 4. «==», «!=». <p>Класс Word<Symbol> должен иметь конструктор без параметров, который создаёт пустое «слово».</p>	Сырбу ИУ9-23 11.05

Таблица 25: Варианты шаблонов классов

№	Описание	Студент
61	<p>Queue<T> – очередь значений типа T, реализованная через кольцевой буфер.</p> <p>Операции, перегружаемые для Queue<T>:</p> <ol style="list-style-type: none"> 1. «<<» – добавление значения в очередь; 2. «>>» – вытаскивание значения из очереди; 3. «length» – получение количества значений в очереди; 4. «[]» – получение ссылки на i-тое значение в очереди; 5. «+» – конкатенация двух очередей, в результате которой формируется новая очередь. 	Чигвинцев ИУ9-23 11.05
62	<p>PolyEq<N> – равенство вида $a_1x^N + a_2x^{N-1} + \dots + a_Nx + a_0 = 0$, в котором a_i и x_i – простые дроби.</p> <p>Операции, перегружаемые для PolyEq<N>:</p> <ol style="list-style-type: none"> 1. «*» – умножение левой и правой части на число; 2. «+» – сложение двух равенств; 3. «/» – деление равенства на PolyEq<1>; 4. «()» – проверка справедливости равенства для указанного значения переменной x. 	Чурсина ИУ9-23 11.05
63	<p>SparseSet<A, B> – разреженное множество с целочисленными элементами, принадлежащими диапазону от A до B, реализованное через два массива sparse и dense.</p> <p>Операции, которые должны быть перегружены для SparseSet<A, B>:</p> <ol style="list-style-type: none"> 1. «+», «*» – объединение и пересечение двух множеств; 2. «<<» – добавление числа в множество (возвращает ссылку на текущее множество); 3. «>>» – удаление числа из множества (возвращает ссылку на текущее множество); 4. «()» – проверка принадлежности числа множеству (принимает число в качестве параметра, возвращает bool); 5. «==», «!=». 	Шевляков ИУ9-23 11.05

Таблица 26: Варианты шаблонов классов

№	Описание	Студент
64	<p>DigitSet<D> – подмножество цифр, используемых в системе счисления по основанию D.</p> <p>Операции, перегружаемые для DigitSet<D>:</p> <ol style="list-style-type: none"> 1. «+», «*» – объединение и пересечение двух множеств, соответственно (одним из операндов этих операций может быть целое число, которое понимается как множество, содержащее цифры представления этого числа в системе счисления по основанию D); 2. «()» – проверка принадлежности указанной цифры множеству; 3. «<<» – вывод цифр множества в поток вывода в порядке возрастания. 	<p>Ширяева ИУ9-23 11.05</p>
65	<p>Curve<T> – кривая на плоскости, заданная функцией $y = f(x)$, где x и y – числа с плавающей точкой типа T. Конструктор кривой принимает в качестве параметра число a и порождает кривую $y = \frac{a}{x}$.</p> <p>Операции:</p> <ol style="list-style-type: none"> 1. «+» – сумма двух кривых: $y = y_1 + y_2 = f_1(x) + f_2(x)$; 2. «-» – разность двух кривых: $y = y_1 - y_2 = f_1(x) - f_2(x)$; 3. «*» – умножение кривой на число: $y = ky_1 = kf_1(x)$; 4. «()» – вычисление определённого интеграла $\int_a^b f(x) dx$. 	<p>Яушев ИУ9-23 11.05</p>