Приложение 2 к РПД
Языки и технологии программирования
01.03.02 Информатика и вычислительная техника
Направленность (профиль)
Технологии разработки мобильных приложений
Форма обучения – очная
Год набора – 2021

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

1.	Кафедра	Математики, физики и информационных технологий
2.	Направление подготовки	09.03.01 Информатика и вычислительная техника
3.	Направленность (профиль)	Технологии разработки мобильных приложений
4.	Дисциплина (модуль)	Б1.О.14.02 Языки и технологии программирования
5.	Форма обучения	очная
6.	Год набора	2021

1. Перечень компетенций

- **ОПК-8:** Способен разрабатывать алгоритмы и программы, пригодные для практического применения

2. Критерии и показатели оценивания компетенций на различных этапах их формирования

Этапы формирования компетенций (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
1. Парадигмы, языки и технологии программирования	ОПК-8	– программирования и реализующие их конкретные технологии; – основные характеристики языка программирования, определения алфавита, синтаксиса и семантики; – основные этапы эволюции языков и технологий программирования; – понятие и состав среды разработки; – способы человеко-машинного взаимодействия при решении задач на ЭВМ; – этапы решения задач на ЭВМ и жизненного цикла программного продукта;	– анализировать условие задачи на целесообразность применения той или иной технологии программирования; – выделять подзадачи в соответствии с выбранной технологией;	– терминологией для определения и описания этапов жизненного цикла программ; – навыками выделения и формулирования этапов решения задач на ЭВМ в контексте выбранной технологии программирования;	Контрольное тестирование №1
2. Разработка программ и реализация основных алгоритмических конструкций средствами выбранного языка программирования и среды разработки.	ОПК-8	–виды модулей в составе проекта (совокупности файлов, обеспечивающих решение задачи на ЭВМ); –процессы, происходящие при сборке проекта, и реализующие их утилиты; –понятия отладки, тестирования, верификации и валидации программ, виды ошибок; –критерии оценки качества программы; –понятие спецификации программы; –определение алгоритма и его свойства, способы записи алгоритма, виды структур алгоритмов; –назначение и правила оформления основных алгоритмических конструкций процедурного программирования; –основные типы данных, их назначение, ограничения, допустимые операции; –способы организации массивов; –способы определения подпрограмм и передачи им параметров, ограничения на типы возвращаемых значений;	– определять, какие действия, на каком этапе разработки программы выполняются; – выделять при решении задачи наиболее важные критерии качества программы; – осуществлять отладку программ различными способами; – выделять и обрабатывать исключительные ситуации (ошибки) вычислительного процесса; – составлять план тестирования, формировать и документировать тестовые наборы; – представлять алгоритмы различными способами; – оценивать вычислительную (временную) и объёмную (пространственную) сложность алгоритмов; – осуществлять программную реализацию алгоритмов; – выполнять рефакторинг, оптимизацию производительности, реинжиниринг уже	– технологией реализации процедурной (структурной) парадигмы программирования средствами выбранной языка программирования; навыками сборки, – отладки, тестирования и документирования программ	Контрольное тестирование №2, лабораторные работы №№ 1-4

Этапы формирования компетенций (разделы, темы дисциплины)	Формируемая компетенция	Критерии и показатели оценивания компетенций			Формы контроля сформированности компетенций
		Знать:	Уметь:	Владеть:	
			существующей программы; –		
3. Обработка отдельных типов данных средствами выбранного языка программирования и среды разработки.	ОПК-8	– приведение и вывод базовых типов данных; – модель работы с выделяемой для программы памятью: статический раздел, стек, куча; – динамическое выделение и освобождение памяти; – механизмы ссылок и указателей; – агрегатные (контейнерные) пользовательские типы данных (перечисления, структуры, массивы, строки, различные виды списков, словари, хэш-таблицы); – стандартные алгоритмы работы с агрегатными типами; операции для работы с файлами входных и выходных данных (текстовыми и бинарными); – модули стандартных библиотек для выполнения операций структурного программирования; – примеры сторонних (инструментальных) библиотек и задач, которые могут быть решены с их помощью средствами языка программирования. –	– осуществлять работу с базовыми и пользовательскими типами данных; – реализовывать алгоритмы чтения, обработки, записи данных во внешний файл; – создавать многомодульные проекты, подключать модули стандартной библиотеки, подключать сторонние библиотеки.	– навыками работы с отдельными типами данных	Защита лабораторных работ №№ 5-8

Шкала оценивания в рамках балльно-рейтинговой системы

«неудовлетворительно» – 60 баллов и менее;

«удовлетворительно» – 61-80 баллов

«хорошо» – 81-90 баллов

«отлично» – 91-100 баллов

3. Критерии и шкалы оценивания

Контрольное тестирование

Процент правильных ответов	0-20	21-40	41-60	61-70	71-80	81-90	91-100
Количество баллов за решенный тест	0	1	2	3	4	5	6

Лабораторная работа
(с оформлением отчёта и устной защитой)

Критерии оценки	*Баллы*
Структура отчёта соответствует формальным требованиям, в нём содержатся достаточно подробные сведения в соответствии с разделом *Порядок выполнения работы*	1
В отчёте представлены краткие ответы на не менее, чем две трети вопросов для самоконтроля	1
Раздел тестирования соответствует требуемому формату и содержанию	1
Демонстрация работы программы: 1) верно обработаны тестовые данные из раздела тестирования; 2) имеет место проверка корректности входных данных; 3) если при обработке корректных данных предполагается возможность исключительной ситуации, она должна быть обработана; 4) отсутствуют неэффективные по используемой памяти или количеству операций конструкций или они устраняются не позже дня защиты; 5) все файлы проекта доступны для проверки Выполнено от 0 до 3 пунктов – 1 балл, более 3-х пунктов – 2 балла.	2
Верные ответы на вопросы преподавателя по содержанию выполненной работы и вопросы для самоконтроля (не более 3-х)	1
Всего	*6*

Типовые контрольные задания и методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

1. Типовое контрольное тестирование (на примере языка программирования C++)

Контрольное тестирование по разделу 2 (по версии языка C++14, среда разработки Qt Creator)

1. Каким будет результат запуска утилиты qmake без параметров в директории проекта?
   1) будет создан файл проекта .pro;
   2) будет создан Makefile;
   3) будет выполнен скрипт в имеющемся Makefile;
   4) будет создан исполняемый модуль.

2. Какая утилита используется в Qt для компиляции файлов ресурсов?
   1) cpp;
   2) uic;
   3) rcc;
   4) moc.

3. Какое словосочетание не относятся к описанию процесса отладки?
   1) стражи включения;
   2) точки останова;
   3) пошаговое выполнение;
   4) дамп памяти.

4. Какой из перечисленных терминов означает процесс разбора символьной последовательности по шаблону с целью поиска и извлечения определённой информации?
   1) сборка;
   2) аллоцирование;
   3) инициализация;
   4) парсинг.

5. Что НЕ может содержаться в заголовочном файле (укажите ВСЕ варианты)?
   1) директивы включения;
   2) директивы условной компиляции;
   3) макроопределения;
   4) объявления типов;
   5) объявления функций;
   6) определения встраиваемых функций;
   7) объявление функции main;
   8) определение функции main;
   9) объявления констант;
   10) определения пространств имён.

6. Как можно назвать функцию main, говоря о процессе передачи управления от операционной системы к программе, написанной на языке C++?
   1) контрольная точка;
   2) точка экстремума;
   3) точка останова;
   4) точка входа.

7. Какая из перечисленных инструкций недопустима в версиях языка, предшествующих C++14?
   1) int a = 1;
   2) int a(1);
   3) int a{1};
   4) все инструкции допустимы.

8. Какая из перечисленных инструкций приведёт к ошибке компиляции в случае версии языка C++14?
   1) int a = 0.5;
   2) int a(0.5);
   3) int a{0.5};
   4) ни одна из перечисленных.

9. Какие из следующих высказываний тождественно истинны (перечислите ВСЕ варианты) с учётом следующего определения: bool b(10):
   1) b == true;
   2) b == false;
   3) b == 10;
   4) b == 1;

10. Какой из побитовых операторов позволяет получить целую часть от деления целого числа на 2 без использования других операторов?
    1) >>;
    2) &;
    3) |;
    4) <<.
11. Какой из следующих циклов является в C++ циклом с постусловием?
    1) for;
    2) foreach;
    3) do-while;
    4) while.

12. Какая подпрограмма на языке C++ наиболее точно соответствует процедуре языка Pascal?
    1) встраиваемая функция;
    2) функция, заданная макроопределением;
    3) функция со спецификатором static;
    4) функция с типом возвращаемого значения void;
    5) функция с пустым списком входных параметров.

13. В программе определены две функции:
    int sum1 (int a, int b) {return a + b;}
    constexpr int sum 2(int a, int b){return a + b;}
    Какая из перечисленных инструкций приведёт к ошибке компиляции:
    1) int a1(sum2 (5, 12));
    2) constexpr int a2(sum1 (5, 12));
    3) int a3(sum2 (5, 12));
    4) int a4(sum1 (5, 12));
    5) ни одна из инструкций не приведёт к ошибке.

14. Что такое rvalue?
    1) именованный объект;
    2) значение, доступное только в пределах выражения, где оно было использовано как литерал или вычислено;
    3) локальная копия переданного в функцию значения ей параметра;
    4) нулевой указатель.

15. Какой из перечисленных операторов имеет наивысший приоритет:
    1) постфиксный инкремент/декремент;
    2) префиксный инкремент/декремент;
    3) взятие адреса;
    4) унарный плюс/минус.

16. Имеется массив *a* (int a[5]), целочисленная переменная i и указатель p: int* p(a). Какие из перечисленных высказываний тождественно истинны (перечислите ВСЕ варианты)?
    1) p == a;
    2) p == &a[1];
    3) a[i] == *(p + i);
    4) a[i] == *(a + i).

17. Какие из следующих типов могут быть использованы для возвращаемого значения функции (перечислите ВСЕ варианты)
    1) void;
    2) int;
    3) int*;
    4) int**;
    5) int&;
    6) int[];
    7) int[5].

18. В какой из перечисленных задач целесообразно использовать статическую переменную?
    1) подсчёт количества вызовов функции;
    2) выделение памяти фиксированного размера вне зависимости от компилятора;
    3) организация цикла со счётчиком;
    4) просмотр значений нестатических переменных в режиме отладки.

19. Что такое рекурсия?
    1) инициализация значением по умолчанию;
    2) исключительная ситуация, приводящая к завершению работы программы;
    3) вычислительный процесс, при котором функция вызывает саму себя;
    4) функция, которая возвращает значение типа void.

20. Что означает рефакторинг?
    1) расширение функциональной спецификации программы;
    2) оптимизация исходного кода по количеству используемой памяти;
    3) оптимизация исходного кода по количеству операций;
    4) преобразование исходного кода с целью сделать его более понятным для программиста.

Ключ:

Номер вопроса	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Ответ	2	3	1	4	8	4	3	3	1,3	1
Номер вопроса	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Ответ	3	4	2	2	1	1,3,4	1,2,3,4,5	1	3	4

2. Типовая задача лабораторной работы (*лабораторная работа №2, язык программирования C++*)

Цель: написать программу, реализующую метод Крамера, Гаусса-Жордана для решения СЛАУ размером (3×3) или отдельные этапы решения в соответствии с этими методами.

Общие входные данные

Входные данные: расширенная матрица СЛАУ (3 строки, 4 столбца) с целочисленными коэффициентами аналогично заданию в лабораторной работе 1. В некоторых задачах достаточно рассматривать только матрицу коэффициентов при неизвестных (то есть 3×3). Чтение осуществляйте в вещественный двумерный массив. Серии из ввода и вывода результатов происходят до тех пор, пока пользователь не введёт символ ‘e’. Предполагается, что элементы матрицы введены корректно (нет ошибок ввода). Форматирование вещественных значений при выводе: один символ после запятой.

Задачи по вариантам

Вариант 8. Вычисление главного определителя СЛАУ по теореме разложения по строке или столбцу с расчётом определителя II порядка по правилу диагоналей.
Дополнительные входные данные. Одна строка, содержащая символ ‘R’ или символ ‘C’ (для строки и столбца соответственно) и номер ряда (данные разделены пробелом).
Выходные данные. Значение определителя.

Порядок выполнения работы

1. Составьте алгоритм решения задачи.
2. Проанализируйте возможные ситуации в решении, включая те, которые не противоречат типам входных данных, но не могут быть обработаны корректно (исключительные ситуации), о чём следует сообщить пользователю (продумайте текст выводимого сообщения). Составьте и запишите тесты для каждой ситуации (всего не менее трёх тестов).
3. Выполните программную реализацию решения задачи.
4. Оцените условное время выполнения (временную сложность) программы, используя профиль для теста, который не содержит исключительную ситуацию, и аналитический профиль. Считайте, что каждая инструкция соответствует одной условной операции. При анализе аналитического профиля подумайте, может ли решение задачи быть обобщено для СЛАУ больших размеров. Выясните, от каких переменных будет зависеть временная сложность. Представьте результаты анализа в виде таблицы и сделайте вывод.
5. Оцените объёмную сложность программы. Представьте результаты анализа в виде таблицы и сделайте вывод.
6. Составьте отчёт, включив в него результаты выполнения предыдущих пунктов.

Материалы отчёта

Алгоритм вычисления главного определителя по теореме разложения по указанному ряду.

1. Организация цикла для решения серии задач пользователя, которая завершается вводом символа ‘e’.
2. Ввод матрицы и параметров для осуществления разложения.
3. Определение некорректных значений: если введён недопустимый символ ряда или номер ряда, вывести сообщение об ошибке и перейти к считыванию данных для новой задачи.
4. Вычислить определитель по теореме разложения, организовав отдельно обработку для случая выбора пользователем строки и для случая выбора столбца.
5. Вывод значения определителя.
6. Переход на пункт 1.

Тестирование программы

Тест 1. Случай решения, когда введен правильно символ строки/столбца и номер строки/столбца

1. Описание входных и выходных данных на языке предметной области.

Входные данные:

$$A = \begin{pmatrix} 2 & 3 & 5 \\ 2 & 5 & 7 \\ 4 & 5 & 3 \end{pmatrix},$$

разложение по строке №1 (нумерация с нуля).

Выходные данные: detA = -24.

2. Анализ и вычисления

$$det\begin{pmatrix} 2 & 3 & 5 \\ 2 & 5 & 7 \\ 4 & 5 & 3 \end{pmatrix} = 2 \cdot \begin{pmatrix} 5 & 7 \\ 5 & 3 \end{pmatrix} \cdot (-1)^{1+1} + 3 \cdot \begin{pmatrix} 2 & 7 \\ 4 & 3 \end{pmatrix} \cdot (-1)^{1+2} + 5 \cdot \begin{pmatrix} 2 & 5 \\ 4 & 5 \end{pmatrix} \cdot (-1)^{1+3}$$
$$= -24$$

3. Таблица входных и выходных данных в требуемом формате

Входные данные:	Выходные данные:

2 3 5 2 5 7 4 5 3 R 1	-24

Тест 2. Случай решения, когда введен правильно символ строки/столбца и неправильно номер строки/столбца.

1. Описание входных и выходных данных на языке предметной области.

Входные данные:

$$A = \begin{pmatrix} 2 & 3 & 5 \\ 2 & 5 & 7 \\ 4 & 5 & 3 \end{pmatrix},$$

разложение по строке №4.

Выходные данные: ошибка в номере ряда.

2. Анализ и вычисления: не требуются.
3. Таблица входных и выходных данных в требуемом формате

Входные данные:	Выходные данные:
2 3 5 2 5 7 4 5 3 R 4	There is no such number

Тест 3. Случай решения, когда введен неправильно символ строки/столбца и правильно номер строки/столбца.

1. Описание входных и выходных данных на языке предметной области.

Входные данные:

$$A = \begin{pmatrix} 2 & 3 & 5 \\ 2 & 5 & 7 \\ 4 & 5 & 3 \end{pmatrix},$$

разложение по ряду №2

Выходные данные: ошибка в символе ряда.

2. Анализ и вычисления: не требуются.
3. Таблица входных и выходных данных в требуемом формате

Входные данные:	Выходные данные:
2 3 5 2 5 7 4 5 3 n 2	There is no such symbol

Оценка временной (вычислительной) сложности программы

Профиль с конкретными входными данными строится по тесту 1. В аналитическом профиле учитываем количество задач *N*, решаемых пользователем до ввода ‘e’. Обобщение на случай матриц большего размера невозможно, так как хотя теорема разложения справедлива для любого порядка, правило диагоналей применяется только для расчёта определителя II порядка. Операторы, ограничивающие область видимости – { и } – не будем учитывать ни в конкретном профиле, ни в аналитическом. Будем считать, что каждой инструкции соответствует ровно одна условная операция (единица времени).

Номер строки	Профиль	Аналитический профиль	Программа
1			#include <iostream>
2			using namespace std;
3			
4			inline int detf(double vals[])
5			{

6			return vals[0]*vals[3] – vals[1]*vals[2];
7			}
8			
9			int main()
10			{
11	1	1	int dr(3);
12	1	1	int dc(3);
13	1	1	double a[dr][dc];
14	1	1	double vals[4]{0.0,0.0,0.0,0.0};
15	1	1	double det(0.0);
16	1	1	int counter(0);
17	1	1	char k;
18	1	1	int number,i,j,m;
19	1	1	double added;
20	1	N	do{
21			
22	1	N	det=0;
23	1	N	counter=0;
24	1	N	k = 'R';
25	1	N	number = 0;
26			
27	3	3N	for (i = 0; i < dr; ++i) {
28	9	9N	for (j = 0; j < dc; ++j) {
28	9	9N	cin >> matr[i][j];
29			}
30	3	3N	cin.get();
31			}
32	1	N	cout << endl;
33	1	N	cin >> k >> number;
34	1	N	cin.get();
35			
36	1	N	if ((k != 'R') && (k !='C'))
37			{
38	0	N/0	cout <<"There is no such symbol"<< endl;
39	0	N/0	continue;
40			}
41			
42	1	N/0	if(number > dr || number < 0)
43			{
44	0	N/0	cout <<"There is no such number"<< endl;
45	0	N/0	continue;
46			}
47	1	N	if (k =='R'){// для строк
48	3	3N/0	for(m = 0, counter = 0;m < dc;m++) {
49	9	9N/0	for (i = 0; i < dr ; ++i) {
50	27	27N/0	for (j = 0; j < dc; ++j) {
51	27	27N/0	if((number != i) && (m != j))
52			{
53	12	27N/0	vals[counter++]=a[i][j];
54			}
55			}
56			}
57	3	3N	det += a[number][m]* detf(vals)*(((number+m)%2)?-1:1);

58			}
59			}
60	0	0/N	else{
61	0	0/3N	for(m = 0, counter = 0;m < dr;m++) {
62	0	0/9N	for (i = 0; i < dr ; ++i) {
63	0	0/27N	for (j = 0; j < dc; ++j) {
64	0	0/27N	if((number != j) && (m != i))
65			{
66	0	0/27N	vals[counter++]= a[i][j];
67			}
68			}
69			}
70	0	0/3N	det += a[m][number]*detf(vals)*(((m+number) %2)?-1:1);
71			}
72			}
73	1	N	printf("%.1f\n\n",det);
74	1	N	}while(cin.get() != 'e');
75	1	1	return 0;
76			}
Всего	119	17*N + 4*3*N + 3*9*N + 3*27*N + 10 = 137N + 10	

Вывод: для выполнения действий для входных данных из теста 1 требуется 119 условных операций; временная (вычислительная) сложность программы в общем случае зависит главным образом от числа решаемых пользователем задач: $C_1(\text{prog}) = 137N + 10 = O(N)$.

Оценка объемной сложности программы

Название подпрограммы	Переменные	Объём, байт
Функция main	dr - int	4 байта
	dc - int	4 байта
	det- double	8 байт
	counter- int	4 байта
	number- int	4 байта
	j - int	4 байта
	i- int	4 байта
	m- int	4 байта
	k - char	1 байт
	vals[4]- double[4]	4*8 = 32 байта
	a[dr][dc]-double[9]	9*8 = 72 байта
Функция detf	Локальные переменные при вызове не создаются, так как функция является встраиваемой.	
	Всего:	141 байт

Вывод: объемная сложность программы составляет 141 байт; размер необходимой области памяти не зависит от количества задач, решаемых пользователем: $C_2(\text{prog}) = O(N^0) = O(1)$.

Подготовка доклада:

Критерии оценивания текста доклада	0-3 балла
Выполнены все требования к содержательной и оформительской части доклада: – текст доклада соответствует теме, тема раскрыта достаточно полно, сделаны необходимые выводы и обобщения, теоретические сведения	3

Критерии оценивания текста доклада	0-3 балла
проиллюстрированы примерами – доклад оформлен в соответствии с требованиями к оформлению – при подготовке доклада использовано не менее трех источников	
При оформлении текста доклада допущены недочеты, не влияющие на его содержательную часть	2
Оценка выставляется, если: – тема доклада раскрыта слабо или неполно – в тексте отсутствуют выводы, обобщения, приведены частные примеры – оформление текста не соответствует требованиям	1
Оценка выставляется, если: – текст доклада не представлен – тема доклада не раскрыта, либо из текста можно сделать вывод о том, что студент не разобрался в материале – текст в значительной мере заимствован из одного или нескольких источников – оформление текста не соответствует требованиям	0
Критерии оценивания выступления	0-2 балла
Выполнены все требования к публичной защите доклада: – во время выступления использованы наглядные материалы (презентация, иллюстрации, схемы) – ответы на уточняющие вопросы демонстрируют понимание студентом темы, аргументированы и подкреплены как теоретическими сведениями, так и практическими примерами	2
Ответы на вопросы неполны либо отсутствуют	1
Выступления нет либо оно проведено неудовлетворительно	0

Требования к докладу

Требования к оформлению доклада:

1. Объем доклада – 5 страниц (без титульного листа и списка источников).
2. Титульный лист должен быть оформлен по образцу (имеется файл с образцом).
3. Основной текст работы оформлен в соответствии с требованиями, указанными ниже.
4. В случае использования в тексте таблиц и/или рисунков на каждый объект должна быть ссылка в тексте работы. Например, «... основные виды программных средств представлены ниже (см. Таблица 1)» или «... схему передачи информации можно увидеть на рис. 1».
5. Количество источников должно быть не менее трех, на все должны быть ссылки внутри текста.
6. Список используемых источников должен быть оформлен в соответствии с требованиями, указанными ниже.

Для оформления основного текста работы:

1. Шрифт – TimesNewRoman, размер – 14 пт.
2. Абзац: междустрочный интервал – 1,5; выравнивание – «по ширине»; абзацный отступ – 1,25 см.
3. Оформление рисунков (при необходимости): выравнивание рисунка – «по центру», подпись рисунка – «Рис. №. Название рисунка»; шрифт для подписи рисунка – TimesNewRoman, размер – 12 пт.
4. Оформление таблиц (при необходимости): выравнивание таблицы – «по центру»; шрифт внутри таблицы – TimesNewRoman, размер – 11-12 пт.; выравнивание текста внутри таблицы – на усмотрение пользователя; подпись таблицы располагается над таблицей и состоит из двух частей: «Таблица №» – выравнивание по правому краю и «Название таблицы» – выравнивание по правому краю или по центру.

Для оформления источников (в соответствии с ГОСТ 2008):

1. Источники должны быть расположены в алфавитном порядке и пронумерованы.
2. В тексте доклада ссылка на источник выполняется в виде: [№], где № – номер источника в общем списке.
3. Если в тексте используется дословная цитата, то она должна быть взята в кавычки, а в ссылке на источник указана страница: [5, с.15].

Примерные темы докладов:

1. Среды разработки, поддерживающие C/C++.
2. Новые возможности языка C++ в стандарте C++11.
3. Новые возможности языка C++ в стандарте C++14.
4. Новые возможности языка C++ в стандарте C++17.
5. Сравнение производительности программ, написанных на C++ и других языках.
6. C++ и Python: какой язык стоит изучать раньше?
7. Умные указатели.
8. Абстракции в манипулировании функциями: указатель на функцию, шаблон функции, лямбда-выражение.
9. Основные возможности библиотеки Boost.
10. Решение задач линейной алгебры средствами библиотеки Boost.

Вопросы к зачету:

1. Язык программирования: семантика, синтаксис, алфавит, идентификатор, объект, представление, переменная, константа, литерал.
2. Временная (вычислительная) и пространственная сложность алгоритма (программы). Жизненный цикл программы.
3. Способы человеко-машинного взаимодействия: терминал и графический интерфейс.
4. Компоненты сред разработки.
5. Императивное программирование: структурное, процедурное, объектно-ориентированное, агентно-ориентированное, обобщённое.
6. Декларативное программирование: функциональное, логическое.
7. Прикладные математические пакеты (среды математического программирования).
8. Краткая история развития языков программирования (упомянуть не менее 10 языков).
9. Среды исполнения JRE и .NET CLR.
10. Модули в составе проекта: исходного текста, заголовочные, объектные, исполняемые.
11. Проект в Qt: утилиты сборки, препроцессор, компилятор, специализированные компиляторы в Qt. Директивы препроцессора.
12. Работа с отладчиком в Qt.
13. Стандартная библиотека языка C++ и её заголовочные файлы. Операторы ввода/вывода. Функция main. Встраиваемые функции. Макроопределения для функций.
14. Идентификаторы для переменных и функций: рекомендации по названиям (Camel Case), объявления, определения, инициализация.
15. Синтаксис C как основа синтаксиса языков C++, Java, C# (сходства и различия).
16. Логический, символьный, целочисленные, вещественные типы: множества значений и допустимые операции. Типы с фиксированным размером. Примеры литералов для различных типов.
17. Инструкции выбора (ветвления) и цикла. Инструкции break и continue. Понятие оператора. Приоритет операторов.
18. Объявление и определение функции. Способы передачи аргументов в функцию. Объявление статических переменных в теле функции. Рекурсия.
19. Пользовательские типы данных: структуры, перечисления. Понятия абстрактный тип данных и структура данных.
20. Явное и неявное приведение базовых типов. Вывод типов. Механизм указателей и ссылок. Адресная арифметика.
21. Структура памяти, выделяемой приложению: статический раздел, стек, куча. Способы очистки памяти. Проблема утечки памяти.
22. Указатели при работе с массивами (одномерными и двумерными).
23. Контейнеры STL, реализующие абстрактный тип данных массив.

24. Контейнеры STL для списка и словаря.
25. Инструкции чтения и записи данных для текстовых и бинарных файлов.
26. Организация работы со строками в C++.
27. Манипуляции с функциями: указатель на функцию, лямбда-выражения, шаблон для функции.
28. Основные составные элементы графического интерфейса оконного приложения (приложения QWidget).
29. Сигналы и слоты в приложениях QWidget.
30. Библиотека OpenGL. Классы Qt, реализующие обёртку для OpenGL.