

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Пекаревский Борис Владимирович  
Должность: Проректор по учебной и методической работе  
Дата подписания: 30.05.2022 15:17:33  
Уникальный программный ключ:  
3b89716a1076b80b2c167df0f27c09d01782ba84



**МИНОБРНАУКИ РОССИИ**  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Санкт-Петербургский государственный технологический институт  
(технический университет)»

УТВЕРЖДАЮ  
Проректор по учебной  
и методической работе  
\_\_\_\_\_ Б. В. Пекаревский  
« \_\_\_\_ » \_\_\_\_\_ 2021 г.

**Рабочая программа дисциплины**  
**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ**

Направление подготовки  
**09.03.03 Прикладная информатика**

Направленность программы бакалавриата  
**Прикладная информатика в химии**

Квалификация

**Бакалавр**

Форма обучения

**Очная**

Факультет **информационных технологий и управления**  
Кафедра **систем автоматизированного проектирования и управления**

Санкт-Петербург  
2021

**Б1.О.12**

## ЛИСТ СОГЛАСОВАНИЯ

Должность разработчика	Подпись	Ученое звание, фамилия, инициалы
Доцент		А. С. Разыграев
Доцент		Р. В. Макарук
Доцент		И. Г. Корниенко
Старший преподаватель		А. К. Федин

Рабочая программа дисциплины «Информационные технологии и программирование» обсуждена на заседании кафедры систем автоматизированного проектирования и управления протокол от «31» августа 2021 года №1  
Заведующий кафедрой

Т. Б. Чистякова

Одобрено учебно-методической комиссией факультета информационных технологий и управления протокол от «15» сентября 2021 года №1

Председатель

В. В. Куркина

## СОГЛАСОВАНО

Руководитель направления подготовки «Прикладная информатика»		доцент И. В. Новожилова
Директор библиотеки		Т. Н. Старостенко
Начальник методического отдела учебно-методического управления		М. З. Труханович
Начальник учебно-методического управления		С. Н. Денисенко

## СОДЕРЖАНИЕ

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.....	4
2. Место дисциплины в структуре образовательной программы.....	6
3. Объем дисциплины.....	6
4.2. Занятия лекционного типа.....	8
4.3. Занятия семинарского типа.....	10
4.3.1. Семинары, практические занятия.....	10
4.3.2. Лабораторные занятия.....	10
4.4. Самостоятельная работа обучающихся.....	11
4.5. Темы контрольных работ.....	12
4.6. Тестирование.....	17
5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.....	19
6. Фонд оценочных средств для проведения промежуточной аттестации.....	19
7. Перечень учебных изданий, необходимых для освоения дисциплины.....	20
8. Перечень электронных образовательных ресурсов, необходимых для освоения дисциплины.....	21
9. Методические указания для обучающихся по освоению дисциплины.....	21
10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.....	21
10.1. Информационные технологии.....	21
10.2. Программное обеспечение.....	21
10.3. Базы данных и информационные справочные системы.....	22
11. Материально-техническое обеспечение освоения дисциплины в ходе реализации образовательной программы.....	22
12. Особенности освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья.....	22

Приложения: 1. Фонд оценочных средств для проведения промежуточной аттестации.

Приложения: 2. Шаблон задания на курсовой проект.

**1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.**

В результате для освоения образовательной программы бакалавриата обучающийся должен овладеть следующими результатами обучения по дисциплине:

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения (дескрипторы)
<p><b>ОПК-7</b> Способен разрабатывать алгоритмы и программы, пригодные для практического применения</p>	<p><b>ОПК-7.1</b> Составление алгоритмов, написание программ и отладка кодов на процедурном языке программирования.</p>	<p><b>Знать:</b> об алгоритмах и блок-схемах алгоритмов (ЗН-1); о синтаксисе и семантике языка программирования Си (ЗН-2); представление данных в программе на языке Си (ЗН-3); операторы языка Си (ЗН-4); структурные типы данных в Си (ЗН-5); процедуры и функции в языке Си (ЗН-6); особенности передачи параметров в функции (ЗН-7); классы памяти в Си (ЗН-8); особенности программирования рекурсии (ЗН-9); адресацию динамической памяти (ЗН-10); особенности работы с файлами на языке Си (ЗН-11); об аргументах командной строки и макросах (ЗН-12). <b>Уметь:</b> по заданию разработать алгоритм решения задачи и реализовать программу на процедурном языке программирования (У-1). <b>Владеть:</b> навыками разработки алгоритмов и программ на процедурном языке программирования (Н-1).</p>
	<p><b>ОПК-7.2</b> Составление алгоритмов, написание программ и отладка кодов на объектно-ориентированном языке программирования.</p>	<p><b>Знать:</b> о классах в С++ и механизмах инкапсуляции (ЗН-13); механизмы организации наследования в С++ (ЗН-14); механизмы организации полиморфизма в С++ (ЗН-15); поточные классы ввода-вывода в С++ (ЗН-16). <b>Уметь:</b> по заданию разработать алгоритм решения задачи и реализовать программу на объектно-ориентированном языке программирования</p>

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения (дескрипторы)
		(У-2). <b>Владеть:</b> навыками разработки алгоритмов и программ на объектно-ориентированном языке программирования (Н-2).
<b>ПК-2</b> Способен разрабатывать и адаптировать прикладное программное обеспечение	<b>ПК-2.11</b> Разработка алгоритмов решения поставленных задач.	<b>Знать:</b> об алгоритмах и блок-схемах алгоритмов (ЗН-17). <b>Уметь:</b> по заданию разработать алгоритм решения задачи (У-3). <b>Владеть:</b> навыками разработки алгоритмов (Н-3).
	<b>ПК-2.12</b> Создание программного кода в соответствии с техническим заданием.	<b>Знать:</b> инструментальные средства разработки программ (ЗН-18); основные схемы проектирования программ (ЗН-19). <b>Уметь:</b> разрабатывать программный код в соответствии с заданием (У-4). <b>Владеть:</b> навыками разработки программ в соответствии с заданием (Н-4).

## 2. Место дисциплины в структуре образовательной программы.

Дисциплина «Информационные технологии и программирование» относится к обязательной части Блока 1 «Дисциплины» образовательной программы бакалавриата (Б1.О.12) и изучается на 1 курсе в 1 и 2 семестре.

Студент должен иметь начальные сведения о компьютерах и программировании в объеме школьного курса информатики. Полученные в процессе изучения дисциплины «Информационные технологии и программирование» знания, умения и навыки могут быть использованы при изучении дисциплин «Разработка программных систем», «Математическая логика и теория алгоритмов», «Методы оптимизации», «Методы поддержки принятия решений», «Разработка программных комплексов для исследований в химии и химической технологии», «Лингвистическое и программное обеспечение автоматизированных информационных систем», при прохождении учебной, производственной и преддипломной практики, а также при выполнении выпускной квалификационной работы.

## 3. Объем дисциплины.

Вид учебной работы	Всего, академических часов	
	1 семестр	2 семестр
<b>Общая трудоемкость дисциплины</b> (зачетных единиц/ академических часов)	4/144	4/144
<b>Контактная работа с преподавателем:</b>	<b>76</b>	<b>72</b>
занятия лекционного типа	36	18
занятия семинарского типа, в т.ч.	36	36
семинары, практические занятия (в т.ч. практическая подготовка)	36 (4)	36 (4)
лабораторные работы		
курсовое проектирование (КР или КП)		КП
КСР (КРП)	4	18
другие виды контактной работы		
<b>Самостоятельная работа</b>	<b>32</b>	<b>36</b>
<b>Форма текущего контроля</b> (Кр, реферат, РГР, эссе)	Кр (3 шт) Тестирование	Кр (4 шт)
<b>Форма промежуточной аттестации</b> (КР, КП, зачет, экзамен)	Экзамен (36)	КП, экзамен (36)

#### 4. Содержание дисциплины.

##### 4.1. Разделы дисциплины и виды занятий.

№ п/п	Наименование раздела дисциплины	Занятия лекционного типа, академ. часы	Занятия семинарского типа, академ. часы		Самостоятельная работа, академ. часы	Формируемые компетенции	Формируемые индикаторы
			Семинары и/или практические занятия	Лабораторные работы			
1.	Основы алгоритмизации и программирование с использованием скалярных типов данных	12	12		10	ОПК-7, ПК-2	ОПК-7.1, ПК-2.11, ПК-2.12
2.	Структурные типы данных и модульное программирование	12	12		10	ОПК-7, ПК-2	ОПК-7.1, ПК-2.11, ПК-2.12
3.	Организация данных на внешних носителях и в оперативной памяти	12	12		12	ОПК-7, ПК-2	ОПК-7.1, ПК-2.11, ПК-2.12
	Итого по плану за 1 семестр	36	36		32		
4.	Инкапсуляция. Классы в С++ и средства их построения	5	9		9	ОПК-7, ПК-2	ОПК-7.2, ПК-2.11, ПК-2.12
5.	Наследование	5	9		9	ОПК-7, ПК-2	ОПК-7.2, ПК-2.11, ПК-2.12
6.	Полиморфизм	4	9		9	ОПК-7, ПК-2	ОПК-7.2, ПК-2.11, ПК-2.12
7.	Классы потоков ввода-вывода	4	9		9	ОПК-7, ПК-2	ОПК-7.2, ПК-2.11, ПК-2.12
	Итого по плану за 2 семестр	18	36		36		

#### 4.2. Занятия лекционного типа.

№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы	Инновационная форма
1	<p>Синтаксис и семантика универсального языка программирования высокого уровня.</p> <p>Структура программы. Описание данных, константы и переменные. Типы переменных. Выражения. Оператор присваивания. Процедуры ввода-вывода. Построение вычислительных программ линейной структуры.</p> <p>Основные и дополнительные структурные конструкции управления процессом вычислений и их реализация операторами языка: условной передачи управления, выбора, конструкции циклов. Организация программ разветвленной и циклической структуры на примере решения задач вычислительной математики: приближенное вычисление корня функции, приближенное вычисление суммы сходящегося бесконечного ряда и др.</p>	12	Л, МК
2	<p>Структурные типы данных: массивы, строки и записи (структуры). Программирование с использованием структурных типов данных: обработка массивов, матриц и текстов.</p> <p>Процедуры и функции. Формальные и фактические параметры. Передача параметров по значению и ссылке. Время жизни и видимость переменных. Организация библиотек подпрограмм (модули). Средства создания универсальных подпрограмм. Рекурсия.</p>	12	Л, МК
3	<p>Типы файлов: текстовые файлы и файлы компонентов. Стандартные процедуры и функции для работы с файлами.</p> <p>Адресация оперативной памяти. Указатели и операции над ними. Динамическое распределение памяти. Динамические структуры данных: одно- и двухсвязные списки, очередь, стек, дек, граф, бинарные сортированные деревья.</p>	12	Л, МК
4	<p>Жизненный цикл разработки программ. Этапы разработки. Формализация и алгоритмизация задач. Принципы оформления программного кода. Проверка и отладка программного кода.</p>	1	



№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы	Иновационная форма
4	Технология объектно-ориентированного проектирования (ООП). Разновидности языков ООП. Язык ООП - C++. Парадигмы ООП. Классы объектов. Структура класса. Компонентные данные и методы класса. Конструкторы классов. Конструктор по умолчанию. Конструктор копирования-инициализации. Деструкторы классов. Явный и неявный вызов деструктора. Обращение к компонентам класса. Доступ к компонентам класса. Друзья классов. Адресация компонентов класса. Статические данные и методы класса. Константы классов. Перегрузка методов и операций. Оператор-функция. Динамическое распределение памяти для объектов и данных классов. Параметризация типов данных. Шаблоны классов и функций. Программирование иерархических объектов. Контейнеры классов. Массивы классов.	4	
5	Базовый и производные классы. Спецификация производных классов. Доступ к наследованным компонентам базового класса. Соотношение между базовыми и производными классами. Субкласс и Суперкласс. Последовательность вызова конструкторов и деструкторов для объектов производных классов. Параметризация производных классов. Множественное наследование. Программная реализация множественного наследования. Проблема неоднозначности при множественном наследовании. Виртуальное наследование. Приведение типов объекта. Контейнеры. Композитные классы.	5	
6	Динамическое связывание. Виртуальные функции и расширенная совместимость типов. Виртуальные классы. Виртуальные деструкторы. Чистые виртуальные функции. Абстрактные базовые классы и конкретные классы. Виртуальное описание геометрических объектов.	4	
7	Иерархия классов потоков ввода-вывода. Форматный ввод-вывод. Операторы форматного ввода-вывода. Флаги управления форматом потоков ввода-вывода. Перегрузка операций форматного ввода-вывода. Манипуляторы потоков форматного ввода-вывода. Методы бесформатного ввода-вывода. Конструкторы файловых потоков. Режимы ввода-вывода файловых потоков. Контроль состояния файловых потоков.	4	

### 4.3. Занятия семинарского типа.

#### 4.3.1. Семинары, практические занятия.

№ раздела дисциплины	Наименование темы и краткое содержание занятия	Объем, акад. часы		Инновационная форма
		всего	в т.ч. на практическую подготовку	
1	Разработка алгоритмов решения задач. Схемы алгоритмов. Программирование с использованием ветвлений. Программирование с использованием циклов.	12	1	Групповая дискуссия
2	Одномерные массивы и строки. Двумерные массивы. Классы памяти. Функции.	12	1	Групповая дискуссия
3	Динамическая память. Структуры. Файлы.	12	1	Групповая дискуссия
4	Динамическое конструирование магических квадратов. Перегрузка арифметических и логических операций с обыкновенными дробями и множествами.	9	1	-
5	Композиция стека для анализа скобочных выражений. Наследование параметрических классов для кольцевой очереди.	9	1	-
6	Реализация множественного наследования при стековой сортировке. Косвенная адресация компонентов абстрактных классов бинарных деревьев.	9	1	-
7	Объектно-ориентированная реализация комбинаторных алгоритмов. Разработка манипуляторов потоков ввода-вывода.	9	2	-

#### 4.3.2 Лабораторные занятия.

Учебным планом не предусмотрены.

#### 4.4. Самостоятельная работа обучающихся.

№ раздела дисциплины	Перечень вопросов для самостоятельного изучения	Объем, акад. часы	Форма контроля
1	Типы данных. Описания констант, переменных, типов. Стандартные типы данных: целый, вещественный, символьный, логический, их представление в памяти. Выражения, операции. Массивы, структуры, объединения, указатели и их описание. Стандартные функции. Присваивание. Ввод-вывод данных. Запись базовых структур алгоритма на базовом языке. Условный оператор. Составной оператор. Операторы циклов с предусловием и с постусловием. Цикл с параметром. Программирование структурированных алгоритмов.	10	Контрольная работа №1 Тестирование №1
2	Обработка массивов и строк. Создание программных модулей. Описание подпрограмм. Обращение к подпрограммам и функциям. Способы передачи параметров по ссылке и значению. Использование библиотечных программ. Рекурсия.	10	Контрольная работа №2 Тестирование №2
3	Использование динамической памяти. Данные и алгоритмы. Методы и средства представления и реализации основных структур данных: очередь, стек, дек, множество, граф, дерево, таблица. Файлы.	12	Контрольная работа №3
4	Практическое изучение механизма инкапсуляции. Программирование точечных конфигураций на плоскости	9	Устный опрос, Контрольная работа №4
5	Практическое изучение одиночного наследования. Программирование двухсвязных списков.	9	Устный опрос, Контрольная работа №5
6	Практическое изучение статического полиморфизма. Программирование рациональных дробей.	9	Устный опрос, Контрольная работа №6
7	Практическое изучение потоковых классов. Программирование потоков ввода-вывода.	9	Устный опрос, Контрольная работа №7

#### 4.5 Темы контрольных работ

В качестве примера содержания контрольной работы №1 по 1 разделу «Основы алгоритмизации и программирование с использованием скалярных типов данных», могут быть рекомендованы следующие задачи:

- Разработать программу для вычисления арифметического выражения и вывода полученного результата. Исходные данные ввести с клавиатуры.

Выражение	Исходные данные
$a = \ln(y^{-\sqrt{ x }}) \cdot (\sin(x) + e^{(x+y)})$	$x, y$

- Разработать программу для вычисления выражения и вывода полученного результата. Соответствующие исходные данные ввести с клавиатуры.

Выражение	Исходные данные
$a = \begin{cases} (x+y)^2 - \sqrt{x \cdot y}, & x \cdot y > 0 \\ (x+y)^2 + \sqrt{ x \cdot y }, & x \cdot y < 0 \\ (x+y)^2 + 1, & x \cdot y = 0 \end{cases}$	$x, y$

- Вычислить и вывести на экран таблицу функции  $y=f(x)$  в интервале  $[a, b]$  с шагом  $h$ .

Функция $y=f(x)$
$y = \begin{cases} \frac{1}{(x+1)\sqrt{x^2+1}}, & x > -1 \\ \frac{-(\ln x )^3 + 3(\ln x )^2/2 + 3(\ln x )/2 + \frac{3}{4}}{2x^2}, & x \leq -1 \end{cases}$
$x \in [-3, 0], h = 0.1$

В качестве примера содержания контрольной работы №2 по 2 разделу «Структурные типы данных и модульное программирование», могут быть рекомендованы следующие задачи:

- Найти произведение

$$P = \prod_{i=1}^{30} Z_i$$

$$Z_1 = 0.45;$$

$$Z_2 = 0.17;$$

$$Z_K = 0.5 \sin 2Z_{K-1} - 0.9 \cos 3Z_{K-2}$$

Массивом не пользоваться.

- Даны действительные числа  $a_1, \dots, a_{15}$ .

Получить

$$\bar{a} = \frac{1}{15} \sum_{i=1}^{15} a_i, \quad s = \sqrt{\frac{\sum_{i=1}^{15} (a_i - \bar{a})^2}{14}}$$

- Задана матрица  $Z(5,4)$ . Найти в каждой строке, если там есть отрицательный элемент, среднее арифметическое всех элементов, исключая нулевые и записать эти значения в

массив В. Вывести исходную матрицу Z и массив В.

В качестве примера содержания контрольной работы №3 по 3 разделу «Организация данных на внешних носителях и в оперативной памяти», могут быть рекомендованы следующие задачи:

- Разработать программу перемножения двух матриц А, В размерности  $n \times n$ . Все матрицы размещаются в оперативной памяти динамически, а значение  $n$  вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходные и результирующую матрицы.

- Ввести 8 символов. В символе с наибольшим кодом заменить 3-й бит нулем, а в символе с наименьшим кодом 4-й бит – единицей. Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

- Разработать функцию filter, которая отфильтровывает все цифры из массива символов. Функция filter получает в качестве аргументов указатель на исходный массив символов, его размер, предикат, указатель на отфильтрованный массив и возвращает размер нового (отфильтрованного) массива, оставляя в нём только те элементы, для которых переданный предикат возвращает логическую истину (предикат – функция, которая возвращает истину или ложь). Протестировать разработанную функцию filter.

В качестве примера содержания контрольной работы №4 по 4 разделу «Инкапсуляция. Классы в C++ и средства их построения», могут быть рекомендованы следующие задачи:

- Разработать объектно-ориентированную программу для вычисления декартовых координат вершин правильного шестиугольника с заданной длиной стороны, центр которого совпадает с началом координат, а две стороны параллельны оси X. Длину стороны шестиугольника необходимо передавать программе аргументом командной строки ее вызова, а полученные координаты его вершин отображать в потоке стандартного вывода. Программная реализация вычислений должна быть основана на разработке класса точки с приватными полями ее декартовых координат, публичными методами доступа к ним и конструктором инициализации их значений. Кроме того, в классе необходимо предусмотреть статический метод для конструирования точки по полярным координатам.

- Разработать объектно-ориентированную программу для вычисления декартовых координат вершин правильного шестиугольника с заданной длиной стороны, центр которого совпадает с началом координат, а две стороны параллельны оси Y. Длину стороны шестиугольника необходимо передавать программе аргументом командной строки ее вызова, а полученные координаты его вершин отображать в потоке стандартного вывода. Программная реализация вычислений должна быть основана на разработке класса точки с приватными полями ее декартовых координат, публичными методами доступа к ним и конструктором инициализации их значений. Кроме того, в классе необходимо предусмотреть статический метод для конструирования точки по полярным координатам.

- Разработать объектно-ориентированную программу для вычисления декартовых координат вершин квадрата с заданной длиной стороны и центром в начале координат, диагонали которого отклонены относительно координатных осей на заданный угол. Длину стороны квадрата и угол поворота его диагоналей необходимо передавать программе аргументами командной строки ее вызова, а полученные координаты его вершин отображать в потоке стандартного вывода. Программная реализация вычислений должна быть основана на разработке класса точки с приватными полями для ее декартовых координат, публичными методами доступа к ним и конструктором инициализации их значений. Кроме того, необходимо предусмотреть статический метод для конструирования точки по полярным координатам.

- Разработать объектно-ориентированную программу для вычисления декартовых координат вершин правильного восьмиугольника с заданной длиной стороны и центром в начале координат, у которого четыре вершины лежат на осях координат. Длину стороны этого восьмиугольника необходимо передавать программе аргументом командной строки ее вы-

зова, а полученные координаты его вершин отображать в потоке стандартного вывода. Программная реализация вычислений должна быть основана на разработке класса точки с приватными полями для ее декартовых координат, публичными методами доступа к ним и конструктором инициализации их значений. Кроме того, в этом классе нужно предусмотреть статический метод для конструирования точки по полярным координатам.

- Разработать объектно-ориентированную программу для поиска точки из любого заданного набора точек на плоскости, суммарное расстояние которой до остальных точек этого набора будет минимально. Декартовы координаты всех точек необходимо указать парами целых чисел в аргументах командной строки вызова программы, координаты полученной точки отображать строкой стандартного вывода. В программе нужно реализовать класс точки с приватными полями ее декартовых координат, публичными методами доступа к ним и конструктором инициализации их значений. Вычисление расстояния между парами любых точек должна обеспечивать внешняя функция. Кроме того, в программе необходимо предусмотреть динамическое распределение памяти для всех заданных точек и массива их адресов.

В качестве примера содержания контрольной работы №5 по 5 разделу «Наследование», могут быть рекомендованы следующие задачи:

- Разработать объектно-ориентированную программу, которая в заданной строке символов переставляет каждый символ с максимальным кодом непосредственно перед ближайшим к нему символом с минимальным кодом без изменения взаимного расположения остальных символов. Исходную символьную строку передавать программе через поток стандартного ввода, а результирующую символьную строку отображать через поток стандартного вывода. В обеих строках все символы с максимальным и минимальным кодом должны указывать знаки «^». Разработка программы должна быть основана на использовании абстрактной структуры двунаправленного связанного списка с реализацией операций просмотра, удаления и вставки его элементов с применением механизма наследования, определяющего базовый класс элемента абстрактного списка и использующего производный класс элемента списка символов.

- Разработать объектно-ориентированную программу, которая в заданной строке символов переставляет каждый символ с максимальным кодом непосредственно после ближайшего к нему символа с минимальным кодом без изменения взаимного расположения остальных символов. Исходную символьную строку передавать программе через поток стандартного ввода, результирующую символьную строку отображать через поток стандартного вывода. В обеих строках все символы с максимальным и минимальным кодом должны указывать знаки «^». Разработка программы должна быть основана на использовании абстрактной структуры двунаправленного связанного списка с реализацией операций просмотра, удаления и вставки его элементов с применением механизма наследования, определяющего базовый класс элемента абстрактного списка и использующего производный класс элемента списка символов.

- Разработать объектно-ориентированную программу, которая в заданной строке символов переставляет каждый символ с минимальным кодом непосредственно перед ближайшим к нему символом с максимальным кодом без изменения взаимного расположения остальных символов. Исходную символьную строку передавать программе через поток стандартного ввода, результирующую символьную строку отображать через поток стандартного вывода. В обеих строках все символы с максимальным и минимальным кодом должны указывать знаки «^». Разработка программы должна быть основана на использовании абстрактной структуры двунаправленного связанного списка с реализацией операций просмотра, удаления и вставки его элементов с применением механизма наследования, определяющего базовый класс элемента абстрактного списка и использующего производный класс элемента списка символов.

- Разработать объектно-ориентированную программу, которая в заданной строке символов переставляет каждый символ с минимальным максимальным кодом непосредственно по-

сле ближайшего к нему символа с кодом без изменения взаимного расположения остальных символов. Исходную символьную строку передавать программе через поток стандартного ввода, результирующую символьную строку отображать через поток стандартного вывода. В обеих строках все символы с максимальным и минимальным кодами должны указывать знаки «^». Разработка программы должна быть основана на использовании абстрактной структуры двунаправленного связанного списка с реализацией операций просмотра, удаления и вставки его элементов с применением механизма наследования, определяющего базовый класс элемента абстрактного списка и использующего производный класс элемента списка символов.

- Разработать объектно-ориентированную программу, которая в каждой входной символьной строке переставляет символы с максимальным кодом в конец строки без изменения взаимного расположения остальных символов. Все входные символьные строки передавать программе через поток стандартного ввода, результирующие строки символов отображать через поток стандартного вывода. Во всех входных и результирующих строках на каждый символ с максимальным кодом должен указывать знак «^». Разработка программы должна быть основана на использовании абстрактной структуры двунаправленного связанного списка с реализацией операций просмотра, удаления и вставки его элементов с применением механизма наследования, определяющего базовый класс элемента абстрактного списка и использующего производный класс элемента списка символов.

В качестве примера содержания контрольной работы №6 по 6 разделу «Полиморфизм», могут быть рекомендованы следующие задачи:

- Разработать объектно-ориентированную программу преобразования заданной обыкновенной дроби в конечную десятичную дробь. Запись обыкновенной дроби следует передавать программе аргументом командной строки, где записи числителя и знаменателя десятичными цифрами разделены знаком «/». Результат преобразований отображать записью вещественного числа с фиксированной точкой строкой стандартного вывода. Программная реализация этого преобразования должна быть основана на разработке класса обыкновенных дробей с приватными полями данных для целочисленных значений числителя и знаменателя, а также оператором преобразования типа для получения вещественного числа по результату их деления. Конструктор класса должен обеспечивать представление исходной символьной записи обыкновенной дроби парой взаимно простых целочисленных значений ее числителя и знаменателя. Для приведения обыкновенной дроби к несократимому виду необходимо предусмотреть компонентный метод, реализующий алгоритм Евклида.

- Разработать объектно-ориентированную программу для округления заданной обыкновенной дроби до ближайшего целого числа, значение которого не превосходит ее величину. Запись обыкновенной дроби передавать программе аргументом командной строки, где числитель и знаменатель разделены знаком «/». Результат округления следует отображать строкой потока стандартного вывода. Программная реализация округления должна быть основана на разработке класса обыкновенных дробей с приватными полями данных для целочисленных значений числителя и знаменателя, а также оператором преобразования типа, который преобразует результат их деления в целое число. Конструктор класса должен обеспечивать представление исходной символьной записи обыкновенной дроби парой взаимно простых целочисленных значений ее числителя и знаменателя. Для приведения обыкновенной дроби к несократимому виду следует предусмотреть компонентный метод, реализующий алгоритм Евклида.

- Разработать объектно-ориентированную программу для округления заданной обыкновенной дроби до ближайшего целого числа, значение которого должно быть не меньше ее величины. Запись обыкновенной дроби следует передавать программе аргументом командной строки, где числитель и знаменатель разделяет знак «/». Результат округления необходимо отображать строкой потока стандартного вывода. Программная реализация округления должна быть основана на разработке класса обыкновенных дробей с приват-

ными полями данных для целочисленных значений числителя и знаменателя, а также с оператором преобразования типа, который преобразует результат их деления в целое число. Конструктор класса должен обеспечивать представление исходной символьной записи обыкновенной дроби парой взаимно простых целочисленных значений ее числителя и знаменателя. Для приведения обыкновенной дроби к несократимому виду следует предусмотреть компонентный метод, реализующий алгоритм Евклида.

- Разработать объектно-ориентированную программу для деления обыкновенной дроби на целое число. Оба операнда необходимо передавать ей через аргументы командной строки. Результат деления следует отображать строкой потока стандартного вывода в формате обыкновенной дроби. Во всех случаях для записи обыкновенной дроби нужно применять символьный формат, где ее числитель и знаменатель разделены знаком «/». Программная реализация вычислений должна быть основана на разработке класса обыкновенных дробей с приватными полями данных для целочисленных значений числителя и знаменателя, а также осуществляться компонентным методом перегрузки оператора «/». Конструкторы класса должны выполнять преобразования обоих операндов в указанный числовой формат обыкновенных дробей. Для приведения исходной и результирующей обыкновенных дробей к несократимому виду следует предусмотреть компонентный метод, реализующий алгоритм Евклида, и перегрузку оператора присваивания.

- Разработать объектно-ориентированную программу для вычисления суммы обыкновенных дробей, которые необходимо передавать ей через два аргумента командной строки. Результат суммирования следует отображать строкой стандартного вывода в формате обыкновенной дроби. Во всех случаях для записи обыкновенной дроби нужно применять символьный формат, где ее числитель и знаменатель разделены знаком «/». Программная реализация вычислений суммы должна быть основана на разработке класса обыкновенных дробей с приватными полями данных для целочисленных значений числителя и знаменателя, а также осуществляться компонентным методом перегрузки оператора «+». Конструкторы класса должны выполнять преобразования обоих операндов в указанный числовой формат обыкновенных дробей. Для приведения исходных и результирующей обыкновенных дробей к несократимому виду следует предусмотреть компонентный метод, реализующий алгоритм Евклида, и перегрузку оператора присваивания.

В качестве примера содержания контрольной работы №7 по 7 разделу «Классы потоков ввода-вывода», могут быть рекомендованы следующие задачи:

- Разработать объектно-ориентированную программу для измерения размера заданного файла в байтах, кило- и мегабайтах. Имя измеряемого файла передается программе через аргумент командной строки ее вызова. Результат измерения размера файла должно отображать информационное сообщение в потоке протокола стандартной диагностики (cerr). Эта диагностика должна иметь формат записи суммы числа мегабайтов, килобайтов и байтов, в совокупности составляющих размер файла (M + K + B). При разработке программы следует использовать методы позиционирования файловых потоков ввода без чтения данных из них.

- Разработать объектно-ориентированную программу для подсчета количества непустых строк любого заданного текстового файла. При этом пустыми считаются строки, где отсутствуют любые символы, кроме необязательных пробелов и табуляций. Имя текстового файла для подсчета числа непустых строк передается программе через аргумент командной строки ее вызова. Результат подсчета должно отображать информационное сообщение в потоке протокола стандартной диагностики (cerr). При разработке программы необходимо использовать методы бесформатного ввода файловых потоков и исключить ее зависимость от длины строк измеряемого файла.

- Разработать объектно-ориентированную программу для подсчета количества слов в любом заданном текстовом файле. При этом словом считается любая последовательность букв без различия регистра, отделенная от других фрагментов текста символами пробелов, табуляций или знаками пунктуации и перевода строки. Имя текстового файла для подсче-



та числа слов передается программе через аргумент командной строки ее вызова. Результат подсчета должно отображать информационное сообщение в потоке протокола стандартной диагностики (cerr). При разработке программы следует использовать оператор форматного ввода файловых потоков.

- Разработать объектно-ориентированную программу подсчета количества абзацев в любом заданном текстовом файле. При этом считается, что абзацы разделены с помощью пустых строк, не содержащих других символов, кроме необязательных пробелов и табуляций. Имя текстового файла в указанном формате передается программе через аргумент командной строки ее вызова. Результат подсчета числа абзацев в нем должно отображать информационное сообщение через поток протокола стандартной диагностики (cerr). При разработке программы следует использовать методы бесформатного ввода класса файловых потоков, а также исключить ее зависимость от длины абзаца.

- Разработать объектно-ориентированную программу для измерения общего числа единичных разрядов во всех байтах любого заданного файла. Его имя передается программе через аргумент командной строки. Результат измерения должно отображать информационное сообщение в потоке протокола стандартной диагностики (cerr). При разработке программы следует использовать методы бесформатного ввода символов классов файловых потоков. Кроме того, в программе должен быть реализован эффективный вычислительный алгоритм, минимизирующий число сравнений разрядов для каждого байта. При этом необходимо также исключить повторные измерения одинаковых байтов.

#### 4.6. Тестирование

В качестве примера оценочных средств для текущего контроля успеваемости студентов по 1 разделу «Основы алгоритмизации и программирование с использованием скалярных типов данных» представлены следующие варианты заданий:

1) Необходимо выбрать один или несколько вариантов ответа:

- В структурном программировании базовыми структурами являются ... (Укажите не менее двух вариантов ответов):

- объединение;
- сопровождение;
- цикл с предусловием;
- цикл с постусловием;
- ветвление;
- последовательность.

- Операцию «переменной x присвоить меньшее из двух значений: a, b» можно реализовать следующими фрагментами программы на языке C ... (Укажите не менее двух вариантов ответов):

- $x = b; \text{ if } (a < x) \ x = a;$
- $\text{if } (a < b) \ x = a; \text{ else } \ x = b;$
- $x = (a < b) ? a : b;$
- $x = \min(a, b);$

- Операцию присваивания вещественной переменной x ее абсолютного значения (модуля) можно реализовать следующими фрагментами программы на языке C ... (Укажите не менее двух вариантов ответов):

- $\text{float } x; \ x = (1 - 2*(x < 0))*x;$
- $\text{float } x; \ \text{if } (x < 0) \ x = -x;$
- $\text{float } x; \ x = ((x < 0) ? -x : x);$
- $\text{float } x; \ x = \text{abs}(x);$
- $\text{float } x; \ x = \text{fabs}(x);$

2) Необходимо установить последовательность расположения операторов фрагмента программы:

- Установите последовательность расположения операторов фрагмента программы

для решения задачи вывода на экран первых 20 чисел Фибоначчи: 1, 1, 2, 3, 5, 8, 13, 21, ... (Укажите правильную последовательность в предложенной совокупности ответов):

```
· }  
· x1 = x1 + x2;  
· printf(“%d “, x1);  
· for (int i = 1; i <= 18; i++){  
· x1 = x2 = 1;  
· int x1, x2;  
· printf(“%d %d “, x1, x2);  
· x2 = x1 - x2;
```

- Установите последовательность расположения операторов фрагмента программы для решения задачи перестановки в обратном порядке элементов одномерного целочисленного массива A, содержащего 10 элементов. (Укажите правильную последовательность в предложенной совокупности ответов):

```
· int count, i, r;  
· int a[10] = {0,1,2,3,4,5,6,7,8,9};  
· }  
· a[i] = a[count - i];  
· a[count - i] = r;  
· r = a[i];  
· count = 9;  
· for(i = 0; i < count/2; i++){
```

- Установите последовательность расположения операторов фрагмента программы для решения задачи вывода на экран максимального значения в вещественной матрице A размером  $n \times m$  ( $n \leq 50$ ,  $m \leq 100$  – исходные данные). (Укажите правильную последовательность в предложенной совокупности ответов):

```
· printf(“\n %f”, max);  
· for (j = 0; j < 100; j++)  
· for (max = a[0][0], i = 0; i < 50; i++)  
· float a[50][100], max; int i, j;  
· max = a[i][j];  
· if (a[i][j] > max)
```

В качестве примера оценочных средств для текущего контроля успеваемости студентов по 2 разделу «Структурные типы данных и модульное программирование» представлены следующие варианты заданий (Необходимо выбрать один или несколько вариантов ответа):

- Для работы с целочисленной матрицей X размером  $n \times n$  ( $n \leq 50$  – исходное данное) используется двумерный массив `int x[50][50]`. Возможными заголовками функций, позволяющей в матрице X определить наибольшее значение, является ... (Укажите правильную последовательность в предложенной совокупности ответов):

```
· void func (int x[][50], int n, int *max)  
· void func (int x[][50], int n)  
· int func (int x[][50], int n)  
· void func (int x[][50], int n, int max)
```

- Имеется целочисленный массив X из n элементов ( $n \leq 1000$  – исходное данное): `int x[1000]`. Возможными заголовками функции, позволяющей в массиве X определить количество элементов, превышающих заданное значение Z, являются ... (Укажите правильную последовательность в предложенной совокупности ответов):

```
· void func (int x[], int n, int z, int *k)  
· void func (int x[], int n, int z, int k)  
· int func (int x[], int n, int z)  
· int func (int x[], int n)
```

- Дана строка S и два символа C1 и C2. Функция, позволяющая заменить в строке S символ C1 на символ C2 и подсчитать количество замен, может иметь следующие заголовки ... (Укажите правильную последовательность в предложенной совокупности ответов):

- int func (char \*s, char c1, char c2)
- void func (char \*s, char c1, char c2, int \*k)
- void func (char s[], char c1, char c2, int k)
- int func (char s[], char c1, char c2)
- int func (char \*s, char c1, c2)

### **5. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине.**

Методические указания для обучающихся по организации самостоятельной работы по дисциплине, включая перечень тем самостоятельной работы, формы текущего контроля по дисциплине и требования к их выполнению размещены в электронной информационно-образовательной среде СПбГТИ(ТУ) на сайте: <http://media.technolog.edu.ru>

### **6. Фонд оценочных средств для проведения промежуточной аттестации.**

Промежуточная аттестация по дисциплине проводится в форме экзамена в 1 семестре и в виде защиты курсового проекта и экзамена во 2 семестре.

Экзамен предусматривают выборочную проверку освоения предусмотренных элементов компетенций и комплектуются теоретическими вопросами (для проверки знаний).

При сдаче экзамена, студент получает два вопроса из перечня вопросов, время подготовки студента к устному ответу - до 30 мин.

Пример варианта вопросов на экзамене в первом семестре:

Билет №1

1. Понятие алгоритма. Свойства алгоритмов. Блок-схема алгоритма. Базовые алгоритмические структуры.
2. Процедуры и функции. Объявления и определения функций. Формальные и фактические параметры. Передача аргументов функции.

Пример варианта вопросов на экзамене во втором семестре:

Билет №1

1. Классы объектов. Структура класса. Компонентные данные и методы класса.
2. Множественное наследование. Программная реализация множественного наследования.

Результаты освоения дисциплины считаются достигнутыми, если для всех элементов компетенций достигнут пороговый уровень освоения компетенции на данном этапе – оценка «удовлетворительно».

## 7. Перечень учебных изданий, необходимых для освоения дисциплины.

### а) печатные издания:

1. Тенишев, Д. Ш. Лингвистическое и программное обеспечение автоматизированных систем : учебное пособие для вузов / Д. Ш. Тенишев; под ред. Т. Б. Чистяковой ; Министерство образования и науки Российской Федерации, Санкт-Петербургский государственный технологический институт (технический университет), Кафедра систем автоматизированного проектирования и управления. – Санкт-Петербург : ЦОП «Профессия», 2010. – 403 с. – ISBN 978-5-91884-017-7.

2. Чистякова, Т. Б. Программирование на языках высокого уровня. Базовый курс : Учебное пособие / Т. Б. Чистякова, Р. В. Антипин, И. В. Новожилова ; Министерство образования и науки Российской Федерации, Санкт-Петербургский государственный технологический институт (технический университет), Кафедра систем автоматизированного проектирования и управления. – Санкт-Петербург : [б. и.], 2008. – 101 с.

3. Чистякова, Т. Б. Программирование на языках высокого уровня. Контрольные работы : Учебное пособие для заочной формы обучения / Т. Б. Чистякова, Р. В. Антипин, И. В. Новожилова ; Министерство образования и науки Российской Федерации, Санкт-Петербургский государственный технологический институт (технический университет), Кафедра систем автоматизированного проектирования и управления. – Санкт-Петербург : [б. и.], 2008. – 101 с.

4. Программирование на языках высокого уровня. Курсовое проектирование : Учебное пособие / Т. Б. Чистякова, Р. В. Антипин, И. В. Новожилова, Л. В. Гольцева ; Министерство образования и науки Российской Федерации, Санкт-Петербургский государственный технологический институт (технический университет), Кафедра систем автоматизированного проектирования и управления. – Санкт-Петербург : [б. и.], 2009. – 38 с.

### б) электронные учебные издания:

5. Программирование. Сборник задач : учебное пособие / О. Г. Архипов, В. С. Батасова, П. В. Гречкина [и др.] ; под редакцией М. М. Марана. – Санкт-Петербург : Лань, 2019. – 140 с. – ISBN 978-5-8114-3857-0 // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com> (дата обращения: 30.08.2021). — Режим доступа: по подписке.

6. Барков, И. А. Объектно-ориентированное программирование : учебник / И. А. Барков. – Санкт-Петербург : Лань, 2019. – 700 с. – ISBN 978-5-8114-3586-9 // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com> (дата обращения: 30.08.2021). – Режим доступа: по подписке.

7. Городняя, Л. В. Парадигма программирования : Учебное пособие / Л. В. Городняя. – 2-е изд., стер. – Санкт-Петербург [и др.] : Лань, 2021. – 232 с. – (Учебники для вузов. Специальная литература). – ISBN 978-5-8114-6680-1 // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com> (дата обращения: 30.08.2021). – Режим доступа: по подписке.

8. Конова, Е. А. Алгоритмы и программы. Язык C++ : учебное пособие / Е. А. Конова, Г. А. Поллак. – 6-е изд., стер. – Санкт-Петербург [и др.] : Лань, 2021. – 384 с. – ISBN 978-5-8114-8487-4 // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com> (дата обращения: 30.08.2021). – Режим доступа: по подписке.

9. Свердлов, С. З. Языки программирования и методы трансляции : учебное пособие / С. З. Свердлов. – 3-е изд., стер. – Санкт-Петербург [и др.] : Лань, 2021. – 562 с. – ISBN 978-5-8114-8195-8 // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com> (дата обращения: 30.08.2021). – Режим доступа: по подписке.

10. Солдатенко, И. С. Практическое введение в язык программирования Си : учебное пособие / И. С. Солдатенко, И. В. Попов. – Санкт-Петербург [и др.] : Лань, 2021. – 132 с. – (Учебники для вузов. Специальная литература). – ISBN 978-5-8114-3150-2 : // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com> (дата обращения: 30.08.2021). – Режим доступа: по подписке.

## **8. Перечень электронных образовательных ресурсов, необходимых для освоения дисциплины.**

учебный план, РПД и учебно-методические материалы: <http://media.technolog.edu.ru>

## **9. Методические указания для обучающихся по освоению дисциплины.**

Все виды занятий по дисциплине «Информационные технологии и программирование» проводятся в соответствии с требованиями следующих СТП:

СТП СПбГТИ 040-02. КС УКДВ. Виды учебных занятий. Лекция. Общие требования;

СТО СПбГТИ 018-2014 КС УКДВ. Виды учебных занятий. Семинары и практические занятия. Общие требования к организации и проведению;

СТП СПбГТИ 048-2009. КС УКДВ. Виды учебных занятий. Самостоятельная планируемая работа студентов. Общие требования к организации и проведению.

Планирование времени, необходимого на изучение данной дисциплины, лучше всего осуществлять на весь семестр, предусматривая при этом регулярное повторение пройденного материала.

Основными условиями правильной организации учебного процесса для студентов является:

- плановость в организации учебной работы;
- серьезное отношение к изучению материала;
- постоянный самоконтроль.

На занятия студент должен приходить, имея знания по уже изученному материалу.

## **10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине.**

### **10.1. Информационные технологии.**

В учебном процессе по данной дисциплине предусмотрено использование информационных технологий:

чтение лекций с использованием слайд-презентаций;

взаимодействие с обучающимися посредством электронной информационно-образовательной среды.

### **10.2. Программное обеспечение.**

Операционная система Microsoft Windows 10 (подписка Azure Dev Tools for Teaching Subscription ID 1831112343).

Visual Studio Community (подписка Azure Dev Tools for Teaching Subscription ID 1831112343).

Visual Studio Enterprise (подписка Azure Dev Tools for Teaching Subscription ID 1831112343).

Pelles C (бесплатная лицензия).

Архиватор 7-zip (открытые лицензии (GNU LGPL, BSD 3-clause License, GNU LGPL with unRAR license restriction)).

Moodle (открытая лицензия, GNU GPL v3).

Adobe Acrobat Reader DC (бесплатная лицензия «ADOBE Personal Computer Software License Agreement»).

LibreOffice (открытая лицензия, Mozilla Public License Version 2.0).

Бесплатные веб-браузеры: Google Chrome (Бесплатная некоммерческая лицензия), Mozilla Firefox (Открытая лицензия (Mozilla Public License V2)), Opera (Бесплатная лицензия (Opera EULA)).

### 10.3. Базы данных и информационные справочные системы.

Справочно-поисковая система «Консультант-Плюс»

Веб-страница журнала «Информационные технологии» <http://www.novtex.ru/IT>

Сайты информационных технологий: <http://inftech.webservis.ru>, <http://citforum.ru>

Информационно-аналитический портал «Научная электронная библиотека» <http://elibrary.ru>

Международные мультидисциплинарные аналитические реферативные базы данных научных публикаций <http://webofknowledge.com>, <http://scopus.com>

Электронно-библиотечные системы:

«Электронный читальный зал – БиблиоТех» <https://technolog.bibliotech.ru/>;

«Лань» <https://e.lanbook.com/books/>.

### 11. Материально-техническое обеспечение освоения дисциплины в ходе реализации образовательной программы.

Для проведения занятий по дисциплине на кафедре систем автоматизированного проектирования и управления СПбГТИ(ТУ) имеется необходимая материально-техническая база, соответствующая действующим санитарным и противопожарным правилам и нормам:

Наименование компьютерного класса кафедры	Оборудование
Класс интегрированных систем проектирования и управления химико-технологическими процессами	30 посадочных мест. Учебная мебель, пластиковая доска. Персональные компьютеры (15 шт.): двухядерный процессор Intel Core 2 Duo (2,33 ГГц); ОЗУ 4096 Мб; НЖМД 250 Гб; CD/DVD привод, DVD-RW; видеокарта NVIDIA GeForce 8500 GT; звуковая и сетевая карты, встроенные в материнскую плату. Персональные компьютеры объединены в корпоративную вычислительную сеть кафедры и имеют выход в сеть «Интернет».
Класс информационных и интеллектуальных систем	40 посадочных мест. Учебная мебель, пластиковая доска. Персональные компьютеры (20 шт.): четырехядерный процессор Intel Core i7-920 (2666 МГц), ОЗУ 6 Гб; НЖМД 250 Гб; CD/DVD привод, DVD-RW; видеокарта NVIDIA GeForce GT 220 (1024 Мб); звуковая и сетевая карты, встроенные в материнскую плату. Персональные компьютеры объединены в корпоративную вычислительную сеть кафедры и имеют выход в сеть «Интернет».
Лекционная аудитория	56 посадочных мест. Учебная мебель. Мультимедийный проектор NEC NP41. Ноутбук Asus абj на базе процессора Intel Core Duo T2000. Мультимедийная интерактивная доска ScreenMedia.

### 12. Особенности освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья.

Для инвалидов и лиц с ограниченными возможностями учебные процесс осуществляется в соответствии с Положением об организации учебного процесса для обучения инвалидов и лиц с ограниченными возможностями здоровья СПбГТИ(ТУ), утвержденным ректором 28.08.2014.

**Фонд оценочных средств  
для проведения промежуточной аттестации по  
дисциплине «Информационные технологии и программирование»**

**1. Перечень компетенций и этапов их формирования.**

Индекс компетенции	Содержание	Этап формирования
ОПК-7	<b>Способен разрабатывать алгоритмы и программы, пригодные для практического применения</b>	начальный
ПК-2	<b>Способен разрабатывать и адаптировать прикладное программное обеспечение</b>	начальный

## 2. Показатели и критерии оценивания компетенций на различных этапах их формирования, шкала оценивания

Код и наименование индикатора достижения компетенции	Показатели сформированности (дескрипторы)	Критерий оценивания	Уровни сформированности (описание выраженности дескрипторов)		
			«удовлетворительно» (пороговый)	«хорошо» (средний)	«отлично» (высокий)
ОПК-7.1 Составление алгоритмов, написание и отладка кодов на процедурном языке программирования; отладка и тестирование работоспособности программы.	Рассказывает об алгоритмах и блок-схем алгоритмов (ЗН-1)	Ответ на вопрос №1 к экзамену	Даёт определения алгоритма и блок-схемы алгоритмов. Перечисляет свойства алгоритмов.	Даёт определения алгоритма и блок-схемы алгоритмов. Перечисляет свойства алгоритмов и базовые алгоритмические структуры.	Даёт определения алгоритма и блок-схемы алгоритмов. Перечисляет свойства алгоритмов и базовые алгоритмические структуры. Приводит пример блок-схемы.
	Рассказывает о синтаксисе и семантике языка программирования Си (ЗН-2)	Ответ на вопрос №2 к экзамену	Даёт определения синтаксиса и семантики языка программирования. Перечисляет неточно структуру алфавита языка Си.	Даёт определения синтаксиса и семантики языка программирования. Перечисляет основные элементы алфавита языка Си.	Даёт определения синтаксиса и семантики языка программирования. Перечисляет основные элементы алфавита языка Си. Приводит общую структуру программы на языке Си.
	Рассказывает о представлении данных в программе на языке Си (ЗН-3)	Ответ на вопрос №3 к экзамену	Даёт определения терминам «объявление, определение и инициализация переменной». Перечисляет неточно основные типы данных языка Си и операции над ними.	Даёт определения терминам «объявление, определение и инициализация переменной». Перечисляет основные типы данных языка Си, операции над ними. Приводит пример объявления константы.	Даёт определения терминам «объявление, определение и инициализация переменной». Перечисляет основные типы данных языка Си, операции над ними. Приводит пример объявления константы. Перечисляет правила преобразования типов данных.
	Рассказывает об операторах языка Си (ЗН-4)	Ответ на вопросы №4-8 к экзамену	Перечисляет неточно указанные операторы. Описывает область их использования.	Перечисляет указанные операторы. Описывает область их использования. Приводит примеры их использования.	Перечисляет указанные операторы. Описывает область и особенности их использования. Приводит примеры их использования.



Код и наименование индикатора достижения компетенции	Показатели сформированности (дескрипторы)	Критерий оценивания	Уровни сформированности (описание выраженности дескрипторов)		
			«удовлетворительно» (пороговый)	«хорошо» (средний)	«отлично» (высокий)
	Рассказывает о структурных типах данных в Си (ЗН-5)	Ответ на вопросы №9-10, 17 к экзамену	Даёт неточные определения заданных структурных типов данных. Описывает операции над ними и их элементами.	Даёт определения заданных структурных типов данных. Описывает операции над ними и их элементами.	Даёт определения заданных структурных типов данных. Описывает операции над ними. Приводит пример использования.
	Рассказывает о процедурах и функциях в языке Си (ЗН-6)	Ответ на вопрос №11 к экзамену	Даёт определения процедуры и функции. Описывает синтаксис объявления и определения функций.	Даёт определения процедуры и функции. Описывает синтаксис объявления и определения функций. Рассказывает отличия формальных от фактических параметров.	Даёт определения процедуры и функции. Описывает синтаксис объявления и определения функций. Рассказывает отличия формальных от фактических параметров. Описывает особенности передачи аргументов. Приводит примеры функций.
	Рассказывает об особенностях передачи параметров в функции (ЗН-7)	Ответ на вопросы №12-13 к экзамену	Рассказывает неточно о вариантах передачи заданных параметров в функцию.	Рассказывает о вариантах передачи заданных параметров в функцию и их особенности.	Рассказывает о вариантах передачи заданных параметров в функцию и их особенности. Приводит примеры.
	Рассказывает о классах памяти в Си (ЗН-8)	Ответ на вопрос №14 к экзамену	Рассказывает о классах памяти в Си.	Рассказывает о классах памяти в Си, модулях, структуры модуля.	Рассказывает о классах памяти в Си, модулях, структуры модуля. Приводит примеры.
	Рассказывает об особенностях программирования рекурсии (ЗН-9)	Ответ на вопрос №15 к экзамену	Даёт определения рекурсии. Приводит пример.	Даёт определения рекурсии. Приводит пример. Описывает достоинства и недостатки.	Даёт определения рекурсии. Приводит пример. Описывает достоинства и недостатки. Рассказывает о стеке вызовов.
	Рассказывает об адресации динамической памяти (ЗН-10);	Ответ на вопрос №16 к экзамену	Даёт определение адресу. Описывает операции над ним. Описывает процедуры выделения и освобождения памяти.	Даёт определение адресу. Описывает операции над ним. Описывает процедуры выделения и освобождения памяти.	Даёт определение адресу. Описывает операции над ним. Описывает процедуры выделения и освобождения памяти. Приводит примеры.

Код и наименование индикатора достижения компетенции	Показатели сформированности (дескрипторы)	Критерий оценивания	Уровни сформированности (описание выраженности дескрипторов)		
			«удовлетворительно» (пороговый)	«хорошо» (средний)	«отлично» (высокий)
	Рассказывает об особенностях работы с файлами на языке Си (ЗН-11)	Ответ на вопрос №18 к экзамену	Описывает типы файлов. Основные операции над файлами.	Описывает типы файлов. Операции над файлами. Особенности работы с разными типами файлов.	Описывает типы файлов. Операции над файлами. Особенности работы с разными типами файлов. Примеры работы с файлами.
	Рассказывает об аргументах командной строки и макросах (ЗН-12)	Ответ на вопрос №19 к экзамену	Рассказывает об аргументах командной строки. Области применения. Рассказывает о не менее двух макросах.	Рассказывает об аргументах командной строки. Рассказывает о макросах. Области применения и особенности использования.	Рассказывает об аргументах командной строки. Рассказывает о макросах. Области применения и особенности использования. Примеры использования.
	Продумывает алгоритм решения задачи и формирует программу на процедурном языке программирования (У-1)	Тестирование №1-2	Выполняет с ошибками не более двух заданий	Выполняет с ошибками не более одного задания	Выполняет правильно все задания
	Решает задачи по разработке алгоритмов и программ на процедурном языке программирования (Н-1)	Контрольные работы №1-3	Разрабатывает программы по заданию с незначительными ошибками. Программный код задокументирован с ошибками.	Разрабатывает программы по заданию. Программный код задокументирован с незначительными с ошибками.	Разрабатывает программы по заданию. Программный код задокументирован.
ОПК-7.2 Составление алгоритмов, написание программ и отладка кодов на объектно-ориентированном языке программирования.	Рассказывает о классах в С++ и механизмах инкапсуляции (ЗН-13)	Ответ на вопросы №20-28 к экзамену	Даёт неточные определения заданных элементов классов и механизма инкапсуляции. Описывает неточно особенности их использования.	Даёт определения заданных элементов классов и механизма инкапсуляции. Описывает особенности их использования.	Даёт определения заданных элементов классов и механизма инкапсуляции. Описывает особенности их использования. Приводит примеры использования.
	Рассказывает о механизмах организации наследования в С++ (ЗН-14)	Ответ на вопросы №29-34 к экзамену	Даёт неточные определения заданных элементов механизма наследования в С++. Описывает неточно особенности их использования.	Даёт определения заданных элементов механизма наследования в С++. Описывает особенности их использования.	Даёт определения заданных элементов механизма наследования в С++. Описывает особенности их использования. Приводит примеры использования.

Код и наименование индикатора достижения компетенции	Показатели сформированности (дескрипторы)	Критерий оценивания	Уровни сформированности (описание выраженности дескрипторов)		
			«удовлетворительно» (пороговый)	«хорошо» (средний)	«отлично» (высокий)
	Рассказывает о механизмах организации полиморфизма в С++ (ЗН-15);	Ответ на вопросы №35-42 к экзамену	Даёт неточные определения заданных элементов механизма полиморфизма в С++. Описывает неточно особенности их использования.	Даёт определения заданных элементов механизма полиморфизма в С++. Описывает особенности их использования.	Даёт определения заданных элементов механизма полиморфизма в С++. Описывает особенности их использования. Приводит примеры использования.
	Рассказывает о потоковых классах ввода-вывода в С++ (ЗН-16).	Ответ на вопросы №43-49 к экзамену	Даёт неточные определения заданных элементов потоковых классов ввода-вывода в С++. Описывает неточно особенности их использования. Описывает неточно особенности их использования.	Даёт определения заданных элементов потоковых классов ввода-вывода в С++. Описывает особенности их использования. Описывает неточно особенности их использования.	Даёт определения заданных элементов потоковых классов ввода-вывода в С++. Описывает особенности их использования. Описывает неточно особенности их использования. Приводит примеры использования.
	Продумывает алгоритм решения задачи и формирует программу на объектно-ориентированном языке программирования (У-2).	Контрольные работы №4-7	Разрабатывает программы по заданию с незначительными ошибками. Программный код задокументирован с ошибками.	Разрабатывает программы по заданию. Программный код задокументирован с незначительными с ошибками.	Разрабатывает программы по заданию. Программный код задокументирован.
	Решает задачу по разработки алгоритмов и программ на объектно-ориентированном языке программирования (Н-2).	Курсовой проект	Разрабатывает программу по заданию с незначительными ошибками. Программный код задокументирован с ошибками.	Разрабатывает программу по заданию. Программный код задокументирован с незначительными с ошибками.	Разрабатывает программу по заданию. Программный код задокументирован.
	ПК-2.11 Разработка алгоритмов решения поставленных задач.	Рассказывает об алгоритмах и блок-схем алгоритмов (ЗН-17)	Ответ на вопрос №50 к экзамену	Даёт определения алгоритма и блок-схемы алгоритмов. Перечисляет свойства алгоритмов.	Даёт определения алгоритма и блок-схемы алгоритмов. Перечисляет свойства алгоритмов и базовые алгоритмические структуры.

Код и наименование индикатора достижения компетенции	Показатели сформированности (дескрипторы)	Критерий оценивания	Уровни сформированности (описание выраженности дескрипторов)		
			«удовлетворительно» (пороговый)	«хорошо» (средний)	«отлично» (высокий)
	Продумывает алгоритм решения задачи и формирует блок-схему решения задачи (У-3).	Контрольные работы №1-7	Разрабатывает алгоритм решения задачи с незначительными ошибками. Формирует блок-схему обобщенного алгоритма решения задачи.	Разрабатывает алгоритм решения задачи. Формирует с незначительными ошибками подробную блок-схему алгоритма решения задачи.	Разрабатывает алгоритм решения задачи. Формирует подробную блок-схему алгоритма решения задачи.
	Решает задачу по разработки алгоритма (Н-3).	Курсовой проект	Разрабатывает алгоритм решения задачи с незначительными ошибками. Формирует блок-схему обобщенного алгоритма решения задачи.	Разрабатывает алгоритм решения задачи. Формирует с незначительными ошибками подробную блок-схему алгоритма решения задачи.	Разрабатывает алгоритм решения задачи. Формирует подробную блок-схему алгоритма решения задачи.
ПК-2.12 Создание программного кода в соответствии с техническим заданием.	Рассказывает об инструментальных средствах разработки программ (ЗН-18).	Ответ на вопрос №51 к экзамену	Рассказывает общую информацию об инструментальных средствах разработки программ.	Рассказывает общую информацию об инструментальных средствах разработки программ.	Рассказывает общую информацию об инструментальных средствах разработки программ. Приводит сравнение нескольких инструментальных средств.
	Рассказывает основные схемы проектирования программ (ЗН-19).	Ответ на вопросы №52-53 к экзамену	Описывает область применения заданных схем проектирования. Рассказывает неточно принципы построения.	Описывает область применения заданных схем проектирования. Рассказывает принципы построения.	Описывает область применения заданных схем проектирования. Рассказывает принципы построения. Приводит пример схем.
	Разрабатывает программный код в соответствии с заданием (У-4).	Контрольные работы №1-7	Разрабатывает программу по заданию с незначительными ошибками. Программный код задокументирован с ошибками.	Разрабатывает программу по заданию. Программный код задокументирован с незначительными с ошибками.	Разрабатывает программу по заданию. Программный код задокументирован.

Код и наименование индикатора достижения компетенции	Показатели сформированности (дескрипторы)	Критерий оценивания	Уровни сформированности (описание выраженности дескрипторов)		
			«удовлетворительно» (пороговый)	«хорошо» (средний)	«отлично» (высокий)
	Решает задачу по разработки программ в соответствии с заданием (Н-4).	Курсовой проект	Обосновывает выбор инструментальных средств разработки программного комплекса. Формирует с незначительными ошибками основные схемы проектирования программ (функциональная структура, блок-схема алгоритма, UML диаграмма вариантов использования). Разрабатывает программу по заданию с незначительными ошибками. Программный код задокументирован с ошибками.	Обосновывает выбор инструментальных средств разработки программного комплекса. Формирует основные схемы проектирования программ (функциональная структура, блок-схема алгоритма, UML диаграмма вариантов использования). Разрабатывает программу по заданию с незначительными ошибками. Программный код задокументирован с ошибками.	Обосновывает выбор инструментальных средств разработки программного комплекса. Формирует основные схемы проектирования программ (функциональная структура, блок-схема алгоритма, UML диаграмма вариантов использования). Разрабатывает программу по заданию. Программный код задокументирован.

### 3. Типовые контрольные задания для проведения промежуточной аттестации

#### а) Вопросы для оценки знаний, умений и навыков, сформированных у студента по компетенции ОПК-7:

1. Понятие алгоритма. Свойства алгоритмов. Блок-схема алгоритма. Базовые алгоритмические структуры.
2. Синтаксис и семантика языка программирования. Алфавит языка Си: лексемы, идентификаторы и ключевые слова, знаки операций, литералы и комментарии. Общая структура программы на языке Си.
3. Объявление, определение и инициализация переменной. Константы. Классификация скалярных типов данных, операции над ними. Операции преобразования типов.
4. Основные операторы языка Си: арифметические, логические, сравнения, присваивания. Особенности их использования. Примеры использования унарных, бинарных и тернарных операторов.
5. Побитовые операторы языка Си. Особенности и примеры их использования.
6. Операторы ветвления. Синтаксис операторов, их особенности и примеры использования.
7. Операторы циклов. Синтаксис операторов, их особенности и примеры использования.
8. Неструктурные операторы передачи управления: оператор безусловного перехода `goto`, оператор досрочного завершения `break`, оператор продолжения `continue`. Синтаксис операторов, их особенности и примеры использования.
9. Структурные типы данных: массивы, строки. Описание, операции над массивами и их элементами, примеры использования.
10. Структурные типы данных: записи (структуры). Описание, операции над структурами и их элементами, примеры использования.
11. Процедуры и функции. Объявления и определения функций. Формальные и фактические параметры. Передача аргументов функции.
12. Передача структурных типов данных в качестве аргументов функции. Особенности передачи и примеры.
13. Передача процедур и функций в другие функции через параметры. Особенности передачи и примеры.
14. Классы памяти. Автоматический, регистровый, внешний и статический классы памяти. Модули. Структура модуля. Законы видимости идентификаторов.
15. Рекурсия. Особенности программирования. Достоинства и недостатки. Понятие стека вызовов.
16. Адресация динамической памяти: понятие адреса, операции получения адреса и разыменования. Процедуры получения памяти и освобождения ее.
17. Списковые структуры данных и основные приемы работы с ними: создание элемента, добавление элемента к списку, удаление элемента из списка. Область применения списковых структур данных.
18. Работа с текстовыми и бинарными файлами. Операции над файлами. Примеры.
19. Аргументы командной строки. Препроцессор языка Си.
20. Классы объектов. Структура класса. Компонентные данные и методы класса.
21. Конструкторы классов. Конструктор по умолчанию. Конструктор копирования-инициализации.
22. Деструкторы классов. Явный и неявный вызов деструктора. Виртуальные деструкторы.
23. Доступ к компонентам класса. Друзья классов. Адресация компонентов класса.
24. Адресация компонентов класса. Статические данные и методы класса. Константы классов.
25. Перегрузка методов и операций. Оператор-функция.

26. Параметризация типов данных. Шаблоны классов и функций.
27. Программирование иерархических объектов. Контейнеры классов.
28. Композитные классы. Массивы.
29. Базовый и производные классы. Спецификация производных классов.
30. Доступ к наследованным компонентам базового класса. Соотношение между базовыми и производными классами.
31. Подкласс и суперкласс. Последовательность вызова конструкторов и деструкторов для объектов производных классов.
32. Множественное наследование. Программная реализация множественного наследования.
33. Виртуальное наследование. Приведение типов объекта.
34. Контейнеры. Композитные классы.
35. Полиморфизм при построении иерархии классов.
36. Виртуальные классы. Виртуальные деструкторы.
37. Чистые виртуальные функции. Абстрактные базовые классы и конкретные классы.
38. Особенности работы с динамическими объектами.
39. Особенности работы с динамическими полями. Копирующий конструктор.
40. Дружественные функции и классы.
41. Переопределение операций. Оператор-функция.
42. Шаблоны классов. Шаблоны функций.
43. Иерархия классов потоков ввода-вывода.
44. Операторы форматного ввода-вывода.
45. Флаги управления форматом потоков ввода-вывода.
46. Манипуляторы потоков форматного ввода-вывода.
47. Перегрузка операций форматного ввода-вывода.
48. Методы бесформатного ввода-вывода.
49. Конструкторы файловых потоков. Режимы ввода-вывода файловых потоков.

**б) Вопросы для оценки знаний, умений и навыков, сформированных у студента по компетенции ПК-2:**

50. Понятие алгоритма. Свойства алгоритмов. Блок-схема алгоритма. Базовые алгоритмические структуры.
51. Инструментальные средства разработки. Функциональные возможности. Сравнение нескольких.
52. Основные схемы проектирования программ: структура информационного и алгоритмического обеспечения, UML-диаграммы вариантов использования. Принципы построения. Пример.
53. Основные схемы проектирования программ: структура информационного и алгоритмического обеспечения, UML-диаграммы вариантов использования. Принципы построения. Пример.

При сдаче экзамена, студент получает два вопроса из перечня, приведенного выше. Время подготовки студента к устному ответу на вопросы - до 30 мин.

**4. Темы курсовых проектов:**

1. Основные алгоритмы на графах. Минимальные покрывающие деревья.
2. Кратчайшие пути из одной вершины и для всех пар вершин на графе.
3. Алгоритмы поиска подстрок. Анализ. Сравнение. Описание. Реализация.
4. Разработка класса, управляющего загрузкой ресурсов приложения из ресурсного файла (файлов) с возможностью шифрования.

5. Разработка класса, управляющего загрузкой ресурсов приложения из ресурсного файла (файлов) с возможностью упаковки по алгоритму Хаффмана.
6. Сжатие изображений. JPEG сжатие. Анализ алгоритма и реализации.
7. Сжатие изображений. Рекурсивный (волновой или wavelet) алгоритм. Анализ алгоритма и реализации.
8. Сжатие изображений. Фрактальное сжатие. Анализ алгоритма и реализации.
9. Использование нейронных сетей для адаптивного сжатия информации. Анализ алгоритма и реализации.
10. Сжатие данных методом Шеннона-Фано. Анализ алгоритма и реализации.
11. Программа проведения экзамена по набору тестов. Разработка базы данных для хранения и анализа результатов.
12. Программа, генерирующая тесты по набору вопросов. Ведение базы данных вопросов, анализ результатов тестирования.
13. Запись действий пользователя (клавиатура, мышь). Статистический анализ нажатия клавиш и других действий в реальном времени.
14. Stealth-клиент мониторинга действий пользователя (клавиатура, экран, ввод-вывод).
15. Программа, показывающая объем памяти, занимаемый файлами на диске.
16. Программа, производящая поиск дублирующихся файлов на диске.

**5. Методические материалы для определения процедур оценивания знаний, умений и навыков, характеризующих этапы формирования компетенций.**

Промежуточная аттестация по дисциплине проводится в соответствии с требованиями СТО СПбГТИ(ТУ) 016-2015. КС УКДВ Порядок организации и проведения зачетов и экзаменов.

По дисциплине промежуточная аттестация проводится в первом семестре в форме экзамена и во втором семестре в форме защиты курсового проекта и экзамена, шкала оценивания – балльная («отлично», «хорошо», «удовлетворительно», «неудовлетворительно»).



### Шаблон задания на курсовой проект

Минобрнауки России  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Санкт-Петербургский государственный технологический институт  
(технический университет)»

#### ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

<b>Направление подготовки</b>	09.03.03 Прикладная информатика
<b>Направленность</b>	Прикладная информатика в химии
<b>Факультет</b>	Информационных технологий и управления
<b>Кафедра</b>	Систем автоматизированного проектирования и управления
<b>Учебная дисциплина</b>	Информационные технологии и программирование
<b>Курс</b> 1	<b>Группа</b> <i>Номер группы</i>
<b>Студент</b>	<i>Фамилия Имя Отчество</i>

**Тема:**        *Тема курсового проекта*

**Цель работы:** углубить теоретические знания, необходимые для реализации объектно-ориентированного подхода к построению модели предметной области и декомпозиции прикладной задачи, объектно-ориентированной технологии проектирования программ. Получить практические навыки объектно-ориентированного программирования в объектно-ориентированной инструментальной среде.

**Исходные данные по проекту (основная литература по теме и электронные ресурсы):**

- 1 *Литература по описанию предметной области и объекта исследования.*
- 2 *Литература по разработке программных комплексов.*
- 3 *Литература по инструментальным средствам разработки автоматизированной системы.*
- 4 *Конспект лекций по дисциплине «Информационные технологии и программирование».*
- 5 *Электронные ресурсы (в том числе Интернет-сайты) по предметной области.*

**Перечень вопросов, подлежащих разработке:**

- 1 Аналитический обзор.
  - 1.1 Обзор и анализ предметной области. Сравнительная характеристика существующих систем-аналогов. Обоснование актуальности проекта.
  - 1.2 Общая характеристика и особенности объекта автоматизированной системы.
  - 1.3 Обзор и обоснование выбора инструментальных средств разработки программного комплекса.
- 2 Цель и задачи курсового проекта.
- 3 Технологическая часть.
  - 3.1 Формализованное описание технологического процесса как объекта управления (исследования, проектирования, мониторинга, обработки информации, обучения или др.).

- 3.2 Постановка задачи управления (или др.). Выбор критериев управления.
- 3.3 Разработка функциональной структуры программного комплекса для автоматизированной системы.
- 3.4 Разработка компонентов информационного обеспечения программного комплекса (база данных, база знаний и др.).
- 3.5 Разработка компонентов математического обеспечения программного комплекса.
- 3.6 Разработка структуры интерфейсов для пользователя и администратора (разработчика) программного комплекса.
- 3.7 Описание структур данных и алгоритмов (формат представления данных в памяти и на внешних носителях).
- 3.8 Описание структуры программы (модули, основные функции, классы и т.д.).
- 3.9 Тестирование программного комплекса (на заданном примере).
- 3.10 Оформление документации (пояснительной записки, презентации) по проекту.

**Перечень графического материала:**

- 1 Формализованное описание объекта автоматизированной системы.
- 2 Функциональная структура программного комплекса.
- 3 Структура и характеристика компонентов информационного обеспечения (инфологическая и даталогическая модели данных).
- 4 Блок-схемы алгоритмов и моделей математического обеспечения.
- 5 UML-диаграммы вариантов использования для пользователя и администратора (разработчика) системы.
- 6 Тестовый пример работы программного комплекса.
- 7 Характеристика аппаратного и программного обеспечения.

**Требования к аппаратному и программному обеспечению:**

Аппаратное обеспечение: характеристика аппаратного обеспечения (технологическое оборудование, технические средства автоматизации, ЭВМ, периферийные устройства).

Программное обеспечение: характеристика программного обеспечения (системного, прикладного).

**Дата выдачи задания:**

**Дата предоставления курсового проекта к защите:**

Заведующая кафедрой, проф.

Т. Б. Чистякова

Руководитель, доц.

И. Г. Корниенко

Консультант, ст. преп.

А. К. Федин

Задание принял к выполнению