

**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ
ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ ПРОФСОЮЗОВ»**

Кафедра Информатики и математики
(полное наименование кафедры)

УТВЕРЖДЕН
на заседании кафедры

Протокол №1 от 01.06.2020

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО ДИСЦИПЛИНЕ

Структуры данных в предметной области

(наименование дисциплины)

09.03.03 «Прикладная информатика»

(код наименования направления подготовки /специальности/)

Прикладная информатика в экономике

(направленность/профиль/)

Санкт-Петербург

1. Общие положения

Фонд оценочных средств (ФОС) по дисциплине используется в целях нормирования процедуры оценивания качества подготовки и осуществляет установление соответствия учебных достижений запланированным результатам обучения и требованиям образовательной программы дисциплины. Предметом оценивания являются знания, умения, навыки и (или) опыт деятельности, характеризующие этапы формирования компетенций у обучающихся. Процедуры оценивания применяются в процессе обучения на каждом этапе формирования компетенций посредством определения для отдельных составных частей дисциплины методов контроля – оценочных средств. Основным механизмом оценки качества подготовки и формой контроля учебной работы студентов являются текущий контроль успеваемости и промежуточная аттестация.

1.1. Цель и задачи текущего контроля студентов по дисциплине

Цель текущего контроля – систематическая проверка степени освоения программы 09.03.03 «Прикладная информатика» дисциплины «Структуры данных в предметной области» уровня достижения планируемых результатов обучения - знаний, умений, навыков, в ходе ее изучения при проведении занятий, предусмотренных учебным планом. Задачи текущего контроля:

1. обнаружение и устранение пробелов в освоении учебной дисциплины;
2. своевременное выполнение корректирующих действий по содержанию и организации процесса обучения;
3. определение индивидуального учебного рейтинга студентов;
4. подготовка к промежуточной аттестации.

В течение семестра при изучении дисциплины реализуется традиционная система поэтапного оценивания уровня освоения. За каждый вид учебных действий студенты получают оценку.

1.2. Цель и задачи промежуточной аттестации студентов по дисциплине.

Цель промежуточной аттестации – проверка степени усвоения студентами учебного материала, уровня достижения планируемых результатов обучения и сформированности компетенций на момент завершения изучения дисциплины. Промежуточная аттестация проходит в форме зачёта.

Задачи промежуточной аттестации:

1. определение уровня освоения учебной дисциплины;
2. определение уровня достижения планируемых результатов обучения и сформированности компетенций;
3. соотнесение планируемых результатов обучения с планируемыми результатами освоения образовательной программы в рамках изученной дисциплины.

2. Перечень компетенций с указанием этапов их формирования в процессе освоения дисциплины

Таблица 1.

№ п\п	Контролируемые темы дисциплины	Код формируемой компетенции	Код и наименование индикатора достижения	Наименование оценочного средства
1	Введение	ПК-18	<p>ПК-18.1. – Знать основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов.</p> <p>ПК-18.2 – Уметь разрабатывать алгоритмы, используя изложенные в курсе общие схемы, методы и приёмы построения алгоритмов, выбирая подходящие структуры данных для представления информационных объектов</p> <p>ПК-18.3. – Владеть инструментарием программирования в системе программирования Microsoft Visual Studio</p>	<p>Опрос</p> <p>Задания для самостоятельной работы</p> <p>Задания для самостоятельной работы</p>
2	Данные динамической структуры	ПК-18	<p>ПК-18.1. – Знать основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов.</p> <p>ПК-18.2 – Уметь разрабатывать алгоритмы, используя изложенные в курсе общие схемы, методы и приёмы построения алгоритмов, выбирая подходящие структуры данных для представления информационных объектов</p>	<p>Опрос</p> <p>Задания для самостоятельной работы</p>

			ПК-18.3. – Владеть инструментарием программирования в системе программирования Microsoft Visual Studio	Задания для самостоятельной работы
3	Линейные динамические списки	ПК-18	<p>ПК-18.1. – Знать основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов.</p> <p>ПК-18.2 – Уметь разрабатывать алгоритмы, используя изложенные в курсе общие схемы, методы и приёмы построения алгоритмов, выбирая подходящие структуры данных для представления информационных объектов</p> <p>ПК-18.3. – Владеть инструментарием программирования в системе программирования Microsoft Visual Studio</p>	<p>Опрос</p> <p>Задания для самостоятельной работы</p>
4	Рекурсия. Поиск и сортировка	ПК-18	<p>ПК-18.1. – Знать основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов.</p> <p>ПК-18.2 – Уметь разрабатывать алгоритмы, используя изложенные в курсе общие схемы, методы и приёмы построения алгоритмов, выбирая подходящие структуры данных для представления информационных объектов</p> <p>ПК-18.3. – Владеть инструментарием программирования в системе программирования Microsoft Visual Studio</p>	<p>Опрос</p> <p>Задания для самостоятельной работы</p>

				Задания для самостоятельной работы
5	Древовидные структуры	ПК-18	<p>ПК-18.1. – Знать основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов.</p> <p>ПК-18.2 – Уметь разрабатывать алгоритмы, используя изложенные на лекциях общие схемы, методы и приёмы построения алгоритмов, выбирая подходящие структуры данных для представления информационных объектов</p> <p>ПК-18.3. – Владеть инструментарием программирования в системе программирования Microsoft Visual Studio</p>	<p>Опрос</p> <p>Задания для самостоятельной работы</p> <p>Задания для самостоятельной работы</p>
6	Обзор применения структурированных типов данных	ПК-18	<p>ПК-18.1. – Знать основные методы разработки алгоритмов и программ, структуры данных, используемые для представления типовых информационных объектов.</p> <p>ПК-18.2 – Уметь разрабатывать алгоритмы, используя изложенные в курсе общие схемы, методы и приёмы построения алгоритмов, выбирая подходящие структуры данных для представления информационных объектов</p> <p>ПК-18.3. – Владеть инструментарием программирования в системе программирования Microsoft Visual Studio</p>	<p>Опрос</p> <p>Задания для самостоятельной работы</p>

				Задания для самостоятельной работы
Результат достижения планируемых результатов изучения дисциплины				зачёт

3. Описание показателей и критериев оценивания компетенций

3.1. Критерии оценивания (текущий контроль)

1. Оценка «отлично» выставляется студенту, если студент имеет глубокие знания учебного материала по теме практического задания, в логической последовательности излагает материал; смог ответить на все уточняющие и дополнительные вопросы;
2. Оценка «хорошо» выставляется, если студент показал знание учебного материала, смог ответить почти полностью на все заданные дополнительные и уточняющие вопросы;
3. Оценка «удовлетворительно» выставляется, если студент в целом освоил материал; однако, ответил не на все уточняющие и дополнительные вопросы;
4. Оценка «неудовлетворительно» выставляется студенту, если он имеет существенные пробелы в знаниях основного учебного материала по теме практического задания, который полностью не раскрыл содержание вопросов, не смог ответить на уточняющие и дополнительные вопросы.

3.2. Критерии оценивания (зачет)

Знания, умения, навыки и компетенции студентов оцениваются следующими оценками: «зачтено», «не зачтено».

«Зачтено» выставляется студенту при условии, что студент твердо знает программный материал, грамотно и последовательно его излагает, увязывает с практикой, владеет необходимыми умениями и навыками в выполнении практических заданий и решении задач.

«Не зачтено» выставляется студенту при условии, что студент не знает значительной части основного программного материала, в ответе допускает существенные ошибки, неправильные формулировки, не владеет необходимыми умениями и навыками в выполнении практических заданий и решении задач.

ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ

Вопросы для подготовки к промежуточной аттестации по дисциплине (зачету)

1. Понятие данных и структуры данных.
2. Данные статической и динамической структуры.
3. Простые типы данных.
4. Составные статические структуры данных.
5. Операции над структурами данных: доступ, поиск, вставка, удаление.
6. Разреженные матрицы: основные понятия.
7. Схемы хранения ленточных матриц.
8. Записи, таблицы, строки, множества, их представление в памяти ПК.
9. Основные понятия о данных динамической структуры.
10. Стеки: назначение, определение, описание.
11. Очереди: определение, организация.
12. Дек: определение, структура, состояния.
13. Примеры алгоритмов, использующих стек и очередь.
14. Односвязные линейные списки: определение, использование.
15. Базовые операции над элементами списка.
16. Двусвязный линейный список.
17. Выделение и освобождение динамической памяти.
18. Формирование упорядоченного списка.
19. Рекурсия: понятие и примеры использования.
20. Алгоритмы поиска и сортировки: основные понятия.
21. Алгоритм двоичного поиска.
22. Сортировка методом пузырька.
23. Быстрая сортировка методом Хоара.
24. Обзор методов сортировки и поиска.
25. Древовидные структуры: назначение, основные понятия и терминология.
26. Двоичные деревья.
27. Графическое представление древовидной структуры.
28. Дерево как динамическая структура данных.
29. Алгоритм построения двоичного дерева.
30. Деревья двоичного поиска.
31. Алгоритмы обхода двоичного дерева (основные понятия и обзор).
32. Рекурсивный алгоритм обхода.
33. Не рекурсивные алгоритмы обхода.
34. Алгоритм поиска по дереву.
35. Дерево сортировки.
36. Удаление из дерева.
37. Графы: основные понятия и определения.
38. Матрицы инцидентности и смежности.
39. Алгоритмы на графах: поиски вершин в графах, поиск в глубину.
40. Алгоритм поиска на графах в ширину.

4. Типовые контрольные задания (тесты, рефераты, курсовые работы, кейсы и др.) и методические материалы, процедуры оценивания знаний, умений и навыков ТЕКУЩИЙ КОНТРОЛЬ

Важнейшей формой учебной отчётности студента является **контрольная работа**.

Выполнение контрольной работы является промежуточной формой отчётности по изучаемой дисциплине и преследует цель лишь оценить способность студента к самостоятельному поиску источников, формированию содержания и его письменного изложения по указанной проблеме. Это важная составляющая изучения дисциплины, а также эффективная форма контроля знаний. При заочном обучении она выступает как обязательная, основная форма самостоятельной работы. В контрольной работе (в соответствии с учебным планом) студент обязан самостоятельно глубоко разобраться в изучаемых проблемах, усвоить суть темы, уяснить её содержание и только затем письменно представить свою отчётную работу.

Выполнение контрольной работы является одним из условий допуска студента к сдаче экзамена. Работа должна соответствовать установленным требованиям, то есть в ней должны быть раскрыты все проблемы, определённые темой. Для этого студент обязан самостоятельно проанализировать первоисточники и дать исчерпывающие ответы на вопросы темы. Контрольная работа — серьёзное учебное задание, и чтобы написать её как следует, необходимо использовать те первоисточники и учебные пособия, которые позволяют полнее разобраться в проблеме. Студент должен регулярно работать в университетской и городской библиотеке, вдумчиво конспектировать лекции преподавателей.

При написании контрольной работы следует обращать особое внимание на грамотное использование терминологии. При употреблении впервые тех или иных терминов и понятий следует давать их определения либо в самом тексте, либо в сносках.

Приступая к контрольной работе, требуется сначала ознакомиться с имеющейся литературой по теме, изучить первоисточники и составить план. Здесь, в отличие от курсовой работы, план предполагает рассмотрение одной, причём довольно широкой, проблемы, и он может состоять из двух-трёх вопросов. Минимальное количество первоисточников, привлекаемых для написания курсовой работы — пять наименований.

Как правило, контрольные работы по дисциплине сугубо индивидуальны, то есть их тематика персонифицирована. Однако в отдельных случаях темы контрольных работ могут быть адресованы и сразу нескольким, и группе в целом. Таким приёмом преподаватель выявляет степень усвоения какой-то важной учебной проблемы и определяет необходимость проведения дополнительных занятий по какой-либо теме. В настоящее время широко используется методика компьютерного тестирования знаний студентов по дисциплинам, в результате чего появляется возможность быстро проверять знания по наиболее важным темам и объективно оценивать их. Эта форма также может выступать как вид контрольной работы.

В качестве контрольной работы широко применяется самостоятельное изучение монографического исследования по конкретной, крайне важной проблеме, требующей глубокого рассмотрения. Этот вид работы предполагает не простое знакомство с определённым монографическим исследованием, а детальное его изучение. Для этого студенту важно знать некоторые правила работы с первоисточником, которым для него будет являться монография. Следует выяснить фамилию автора, его имя и отчество, учёную степень и звание, а также что побудило его взяться за изучение данной проблемы; обратить внимание на основные вопросы монографии и их разрешение автором, уметь раскрывать их в ходе собеседования с преподавателем.

Студенту следует письменно (предельно кратко) очертить те вопросы (полностью или частично), которые поставлены автором в монографическом исследовании; при изложении их следует указывать страницы источника.

Задания для написания контрольных работ (для заочной формы обучения)

1. Понятие данных и структуры данных. Классификация типов данных.
2. Составные статические структуры данных.
3. Основные понятия о данных динамической структуры.
4. Стеки: назначение, определение, описание.
5. Очереди: определение, организация.
6. Дек: определение, структура, состояние.
7. Примеры алгоритмов, использующих стек и очередь.
8. Односвязные линейные списки: определение и функциональная спецификация.
9. Базовые операции над элементами списка.
10. Двусвязный линейный список.
11. Выделение и освобождение динамической памяти.
12. Формирование упорядоченного списка.
13. Рекурсия: понятие и примеры использования данных при применении рекурсии.
14. Использование данных в алгоритмах поиска и сортировки (основные понятия).
15. Алгоритм двоичного поиска.
16. Сортировка методом пузырька.
17. Быстрая сортировка методом Хоара.
18. Обзор методов сортировки и поиска.
19. Древовидные структуры: назначение, основные понятия и терминология.
20. Двоичные деревья.
21. Графическое представление древовидной структуры.
22. Дерево как динамическая структура данных.
23. Алгоритм построения двоичного дерева.
24. Деревья двоичного поиска.
25. Алгоритмы обхода двоичного дерева (основные понятия и обзор).
26. Кодирование информации с использованием двоичных деревьев.
27. Рекурсивный алгоритм обхода.
28. Не рекурсивные алгоритмы обхода.
29. Алгоритм поиска по дереву.
30. Дерево сортировки.
31. Удаление из дерева.

Принципы выбора темы работы

Студенты при написании контрольной работы могут выбрать любую из предложенных тем на свое усмотрение.

Требования к оформлению контрольной работы подробно представлены в Положении о бюро контрольных работ, размещенном на сайте Университета в личном кабинете на странице в Системе поддержки самостоятельной работы студентов **ПОЛОЖЕНИЕ О БЮРО КОНТРОЛЬНЫХ РАБОТ _ для работ студентов заочной формы обучения.**

ТЕКУЩИЙ КОНТРОЛЬ

Дискуссионные столы и кейс-задачи в программе не предусмотрены. Текущий контроль осуществляется по результатам выполнения заданий практикума по дисциплине «Структуры данных в предметной области» и заданий для самостоятельной работы.

ПАСПОРТ ТЕСТОВЫХ ЗАДАНИЙ

1. Общее количество тестовых заданий в базе – 100.
2. Ограничение времени выполнения теста (в минутах) – одна попытка, 35 минут.
3. Автоматическое перемешивание вопросов в тесте: - да (нет).
4. Случайный порядок ответов в тестовом задании: - да (нет).
5. Критерии оценки результатов тестирования:

- Неудовлетворительно – 0 –55% правильных ответов.
- Удовлетворительно -55 – 75% правильных ответов.
- Хорошо – 75 -90% правильных ответов
- Отлично – 90% и более правильных ответов

Пример тестовых заданий для текущего контроля представлен ниже:

1. Сколько байтов будет выделено компилятором для хранения следующей структуры:

```
struct d
{
  char c[3];
  int i;
```

- 2;
 - 3;
 - 4;
 - 5;
 - 6;
 - 7.
2. Каким образом можно определить массив указателей на строки:
- char keyword(nkeys);
 - char keyword(*nkeys);
 - char *keyword[nkeys];
 - char keyword.nkeys.
3. Укажите правильный формат записи указателя на структуру date:
- struct date p=d;
 - struct date *p=&d;
 - struct date p=*d;
 - struct date p= =d.
4. Если в программе записано struct node * next, то это означает, что
- задана структура;
 - используется структурная переменная;
 - используется указатель на структуру;
 - используется указатель на массив.
5. С использованием какой команды можно осуществить доступ к элементу структуры:
- st.name = “Петров”;
 - struct node * next ;} st1_node;
 - st1.id = st2.id;
 - struct node { int data; struct node * next; } st1_node;
6. Дано следующее описание переменных:

каких операторов являются неправильными:

```
int *p,*q;
char *r;
```

- p=NULL;
- r=NULL;
- q=*p;
- *p=NULL.

7. Приведите значение переменной r после выполнения следующего фрагмента программы:

```
int main ()
{
  int t=2, b=7, r=3;
  if (t>b)
  {
    if (b < r) r=b;
  }
}
```

- 2;
- 3;
- 4;
- 5;
- 6.

8. Какое значение получит переменная *i* после выполнения оператора case 4:

```
int i=2;
switch (i)
{
case 1: i += 2;
case 2: i *= 3;
case 0: i /= 2;
case 4: i -= 5;
default;
```

- переменная *i* примет значение 4, а затем значение -1;
- переменная *i* примет значение 5, а затем значение -3;
- переменная *i* примет значение 3, а затем значение -2;
- переменная *i* примет значение 6, а затем значение -5.

9. Приведите значение переменной *r* после выполнения следующего фрагмента программы:

```
int main ( )
{
int t=2, b=7, r=3;
if (t>b)
if (b < r) r=b;
else r=t;
return (0);
}
```

- 2;
- 3;
- 4;
- 5;
- 6.

10. Какой оператор отвечает за выбор команды в операторе switch:

- default;
- case;
- break;
- goto;
- return.

11. Чтобы создать простую программу в интегрированной среде из списка типов файлов необходимо выбрать следующий тип файлов:

- Binary File;
- Active Server Page;

- C++ Source File;
- Bitmap File.

12. Какую комбинацию клавиш нужно нажать для компиляции исходного кода программы:

- Shift+F3;
- Ctrl+Alt+Delete;
- Ctrl+F5;
- Alt+F5.