

Государственное образовательное учреждение
«Приднестровский государственный университет им. Т.Г. Шевченко»

Рыбницкий филиал

Кафедра информатики и программной инженерии

«УТВЕРЖДАЮ»

Зав. кафедры-разработчика

 Л.А. Тягульская
протокол № 2 «21» сентября 2023г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО ДИСЦИПЛИНЕ

Б1.О.22. «ПРОГРАММИРОВАНИЕ»

Направление:

09.03.04 «Программная инженерия»

Профиль:

«Разработка программно-информационных систем»

Квалификация:

бакалавр

Форма обучения:

очная, заочная

ГОД НАБОРА: 2023

Разработала: ст. преподаватель

 Гарбузняк Е.С.

« 21 » 09 20 23 г.

Рыбница, 2023

ПАСПОРТ
фонда оценочных средств по учебной дисциплине

1. В результате изучения дисциплины «Программирование» у обучающихся должны быть сформированы следующие компетенции:

Категория (группа) компетенций	Код и наименование	Код и наименование индикатора достижения универсальной компетенции
Общепрофессиональные компетенции и индикаторы их достижения		
	ОПК-1. Способен применять естественнонаучные и общетехнические знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности	ИД-1 _{ОПК-1.1.} Знает основы математики, физики, вычислительной техники и программирования. ИД-2 _{ОПК-1.2.} Умеет решать стандартные профессиональные задачи с применением естественнонаучных и общетехнических знаний, методов математического анализа и моделирования.
	ОПК-6. Способен разрабатывать алгоритмы и программы, пригодные для практического использования, применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	ИД-1 _{ОПК-6.1.} Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды информационных систем и технологий. ИД-2 _{ОПК-6.2.} Умеет применять основные языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ.

2. Программа оценивания контролируемой компетенции:

№ п/п	Контролируемые модули, разделы (темы) дисциплины и их наименование	Код контролируемой компетенции (или ее части)	Наименование оценочного средства
1.	ТЕКУЩИЙ КОНТРОЛЬ 1. Введение. Синтаксис языка C	ОПК-1, ОПК-6. ИД-1 _{ОПК-1.1} ИД-1 _{ОПК-6.1}	Работа на лекциях Практические работы Лабораторные работы Самостоятельная работа

	2. Базовые типы данных. Операции над данными.	ИД-1 _{ОПК-1.1} ИД-1 _{ОПК-6.1}	Работа на лекциях Практические работы Лабораторные работы Самостоятельная работа
	3. Структурное программирование.	ИД-2 _{ОПК-1.2} ИД-1 _{ОПК-6.1}	Работа на лекциях Практические работы Лабораторные работы Самостоятельная работа Модульная контрольная
	4. Модульное программирование	ИД-2 _{ОПК-1.2} ИД-1 _{ОПК-6.1}	Работа на лекциях Практические работы Лабораторные работы Самостоятельная работа Модульная контрольная
	5. Составные типы данных	ИД-2 _{ОПК-6.2}	Работа на лекциях Практические работы Лабораторные работы Самостоятельная работа
	6. Стандартные библиотеки C++	ИД-2 _{ОПК-6.2}	Работа на лекциях Практические работы Лабораторные работы Самостоятельная работа
	7. Введение в объектно-ориентированное программирование	ИД-2 _{ОПК-1.2} ИД-2 _{ОПК-6.2}	Работа на лекциях Практические работы Лабораторные работы Самостоятельная работа
2.	ПРОМЕЖУТОЧНЫЙ КОНТРОЛЬ (1 семестр)	ОПК-1, ОПК-6.	Тестирование Вопросы к экзамену
3.	ИТОГОВЫЙ КОНТРОЛЬ (2 семестр)	ОПК-1, ОПК-6.	Тестирование Вопросы к экзамену Курсовая работа

«УТВЕРЖДАЮ»

зав. кафедрой информатики и
программной инженерии,

доцент _____ Л.А. Тягульская

«___» _____ 2023 г.

**Вопросы и задания для текущего контроля
по дисциплине «Программирование»
для студентов I курса
направления «Программная инженерия»
профиля «Разработка программно-информационных систем»,
1 семестр**

Модульная контрольная работа по разделу «Структурное программирование»

1. Оператор многонаправленного ветвления *switch*.
2. Использование инструкций *break* и *default*.
3. Вложенные инструкции *switch*.

1. Написать программу, которая по номеру дня недели (целому числу от 1 до 7) выдает в качестве результата количество пар и их названия в вашей группе в этот день.

2. Написать программу, позволяющую по последней цифре числа определить последнюю цифру его квадрата.

3. Написать программу, которая по заданному году и номеру месяца *m* определяет количество дней в этом месяце.

4. Для каждой введенной цифры (0-9) вывести соответствующее ей название на английском и русском языке.

5. Написать программу, которая по номеру месяца выдает название следующего за ним месяца. (При *m*=1 получаем- февраль, 4- май и т.д.)

6. Напишите программу, которая анализирует человека по возрасту и относит к одной из четырех групп: дошкольник, ученик, работник, пенсионер. Возраст вводится с клавиатуры.

7. В старояпонском календаре был принят 12-летний цикл.

Годы внутри цикла носили названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. Написать программу, которая вводит номер некоторого года и печатает его название по старояпонскому календарю.

8. Написать программу, которая по введенному номеру времени года (1-зима, 2-весна, 3-лето, 4-осень) выдавала соответствующее этому времени года месяцы и количество дней в каждом месяце.

9. Написать программу, которая по номеру месяца (целому числу от 1 до 12) выдает в качестве результата названия сезона, соответствующее ему. Например: 5-весна.

10. Составьте программу, которая для целого числа *k* (от 1 до 99), введенного вами, напечатает фразу «Мне *k* лет», при этом в нужных случаях слово «лет» заменяя на слово «год» или «года».

Перечень тем рефератов

1. Язык С: история, первоначальная область применения.
2. Изучение инструментальной среды Turbo С.
3. Сравнительный анализ операторов выбора.
4. Сравнительный анализ операторов цикла.
5. Решение задач с использованием различных операторов ввода и вывода.
6. Решение задач с использованием целых типов данных.
7. Решение задач с использованием вещественных типов данных.
8. Решение задач с использованием математических функций.
9. Операции отношения. Логические операции.
10. Решение задач с использованием логических операций.

«УТВЕРЖДАЮ»

зав. кафедрой информатики и
программной инженерии,

доцент *Л.А. Тягульская*

« 21 » 09 2023 г.

**Вопросы и задания для текущего контроля
по дисциплине «Программирование»
для студентов I курса
направления «Программная инженерия»
профиля «Разработка программно-информационных систем»,
2 семестр**

Модульная контрольная работа по разделу «Модульное программирование»

1. Модульное программирование. Функции. Рациональные размер и количество параметров функции. Пример функции.
 2. Функции. Аргументы и параметры. Передача аргументов по значению и по ссылке. Прототипы функций. Преобразование аргументов в точке вызова. Оператор return.
 3. Область действия переменных. Локальные и глобальные переменные. Видимость переменных.
 4. Время жизни переменной. Модификаторы переменных. Изменяющиеся переменные.
 5. Рекурсивные подпрограммы. Понятие о рекурсии. Сравнение рекуррентности и рекурсии.
 6. Прямой и обратный ход рекурсии. Точка останова работы рекурсивного алгоритма. Глубина рекурсии.
- Необходимо реализовать пользовательскую подпрограмму согласно индивидуальному заданию.

Перечень тем рефератов

1. Косвенная рекурсия
2. Создание заголовочных файлов
3. Область видимости переменных. Сравнительная характеристика
4. Время жизни переменных
5. Решение задач с применением пирамидальной и бинарной сортировок
6. Сравнительная характеристика сортировки Шелла и сортировки слиянием
7. Кодировка ASCII и Unicode
8. Выравнивание структур памяти
9. Сравнительная характеристика: различие объединений и структур
10. Сравнительная характеристика: различие режимов открытия файлов
11. Использование операций seek и feof
12. Сохранение и чтение структур из файлов
13. Сравнительная характеристика: различие типа string и стандартных строк
14. Реализация вывода текста в графическом режиме
15. Создание графического меню
16. Изменение цвета и заливки
17. Система координат в графическом режиме
18. Истоки и эволюция C++
19. C++ и реализация объектно-ориентированного программирования
20. Сравнительный анализ языков C и C++
21. Связь C++ с языками Java и C#.

«УТВЕРЖДАЮ»

зав. кафедрой информатики и
программной инженерии,
доцент  Л.А. Тягульская
«21» 09 2023 г.

**Тестовые задания для промежуточного контроля
по итогам освоения дисциплины, а также для контроля самостоятельной работы
студента (1 семестр)**

Структура педагогических измерительных материалов

Номер задания	Наименование темы задания
ДЕ 1. Введение. Синтаксис языка C (критерий освоения ДЕ: не менее 5 правильно выполненных заданий)	
1-5	Среда Borland C.
6-8	Заголовочные файлы. Назначение стандартных заголовочных файлов.
9-12	Ввод и вывод информации. Форматированный вывод.
13,14	Алфавит языка. Способы описания синтаксиса языка. Определение понятия «идентификатор». Служебные слова.
ДЕ 2. Базовые типы данных. Операции над данными (критерий освоения ДЕ: не менее 6 правильно выполненных заданий)	
15-18	Операции отношения. Логические операции. Приоритет операций.
19,20	Вещественные типы данных. Арифметические операции для вещественных операндов. Математические функции стандартной библиотеки Си.
21-32	Целые типы данных. Арифметические операции для целых операндов.
ДЕ 3. Структурное программирование (критерий освоения ДЕ: не менее 11 правильно выполненных заданий)	
33-48	Операторы ветвления: if и switch.
49-65	Операторы цикла: с предусловием, с постусловием и со счетчиком.
66-68	Изменение хода выполнения цикла с помощью операторов break и continue.
ДЕ 4. Модульное программирование (критерий освоения ДЕ: не менее 4 правильно выполненного задания)	
69-80	Функции. Аргументы и параметры. Прототипы функций.

Указания: Напишите Вашу фамилию, номер группы и дату. Для ответа на вопрос с выбором варианта ответа достаточно написать номер вопроса и рядом букву, обозначающую правильный вариант из предложенных в тексте ответов на вопрос. Если Вы считаете правильными несколько вариантов ответов, то запишите через запятую соответствующие литеры букв. Ответ считается верным, если отмечены все правильные варианты ответов.

Время выполнения – 90 минут.

ДЕ 1. Введение. Синтаксис языка C – 14

1. Функциональная клавиша, позволяющая перейти к следующему окну:

- a) F3;
- b) F4;
- c) F5;
- d) F6.

2. Комбинация клавиш, позволяющая зайти в режим Window/Size/Move:

- a) Shift+F5;
- b) Ctrl+F5;
- c) Shift+F6;
- d) Ctrl+F6.

3. Комбинация клавиш, позволяющая откомпилировать программу:

- a) Alt+F9;
- b) Ctrl+F9;
- c) Ctrl+F5;
- d) Alt+F5.

4. Комбинация клавиш, позволяющая стереть строку, на которой располагается курсор:

- a) Ctrl+K;
- b) Ctrl+Y;
- c) Alt+K;
- d) Alt+Y.

5. Комбинация клавиш Shift+Del позволяет:

- a) скопировать;
- b) вырезать;
- c) вставить;
- d) очистить.

6. Директива, позволяющая включить в текст программ другой текстовый файл:

- a) define;
- b) include;
- c) pragma;
- d) void.

7. Правильными высказываниями о директивах препроцессора являются:

- a) в конце директивы должна стоять точка с запятой;
- b) директива начинается со знака #.

8. Функции printf и scanf описываются в библиотечном файле:

- a) iostream.h;
- b) stdio.h;
- c) conio.h;
- d) math.h.

9. Для комментариев могут использоваться символы:

- a) /* comment */;
- b) { comment };
- c) (* comment *).

10. Укажите строки, не содержащие ошибок синтаксиса:

- a) printf("Hello, World!\n");
- b) printf("Hello, World!\n");
- c) printf(%d Hello, \n Hello, World!);
- d) printf("Hello,\n World!\n").

11. В результате выполнения операторов

```
int n; n=5;
printf("%d+%d=%d", n, n, n+n);
```

будет напечатано:

- a) 5 + 5=10;
- b) 5;
- c) 10;
- d) 5+5.

12. Укажите, будет ли работать следующая программа:

```
#include <stdio.h>
int main()
{
    printf("Hello, World!\n");
}
```

- a) программа скомпилируется и отработает, но код завершения будет не определен;
- b) программа не скомпилируется из-за синтаксической ошибки;
- c) программа не скомпилируется, т.к. не указан возвращаемый функцией тип.

13. Элементами алгоритмического языка C являются:

- a) идентификаторы;
- b) ключевые слова;
- c) разделители;
- d) римские цифры.

14. На языке Си правильно написаны имена:

- a) Name;
- b) name_;
- c) +name;
- d) 3name.

ДЕ2 Базовые типы данных. Операции над данными – 18

15. Унарными операциями являются:

- a) &, !, ++;
- b) &, ~, ^;
- c) *, +, %;
- d) >, --, |.

16. Бинарные арифметические операции используются в выражениях:

- a) $X + Y$;
- b) $c \% d + 2$;
- c) $xx * Y$;
- d) $xx++$.

17. Укажите правильный приоритет операций.

- a) $()$, $<<>$, $+$, $!$;
- b) $!$, $<$, $\&\&$, $?:$;
- c) $\&$, $-$, $\|$, $()$;
- d) $()$, $\wedge$, $*$, $+$.

18. Если (Выражение1) в (Выражение1) ? (Выражение2): (Выражение3) будет истинно:

- a) выполнится выражение 2;
- b) выполнится выражение 3;
- c) выполнится выражение 2, а затем выражение 3;
- d) выполнится выражение 3, а затем выражение 2.

19. Математическая функция, вычисляющая $\ln(x)$, выглядит:

- a) $\ln(x)$;
- b) $\log(x)$;
- c) $\lg(x)$;
- d) $\log_e(x)$.

20. Укажите соответствие математической функции и ее назначения:

- a) $\text{abs}(x)$
- b) $\text{sqrt}(x)$
- c) $\text{exp}(x)$
- d) $\text{fabs}(x)$
- 1) Вычисляет «e» в степени x ;
- 2) Возвращает модуль целого числа;
- 3) Возвращает модуль дробного числа;
- 4) Вычисляет квадратный корень.

21. После выполнения операторов $x = 5$; $x += --x$; x будет равен:

- a) 4;
- b) 5;
- c) 8;
- d) 6.

22. В случае $\text{int} t = 11/2$; переменная t примет значение:

- a) 5;
- b) 5.4999;
- c) 5.5.

23. В случае $\text{int} c; c = 11\%3$; переменная c примет значение:

- a) 0.2;
- b) 2;
- c) 3.2.

24. `unsignedchar` принимает диапазон значений:

- a) $[-128..127]$;
- b) $[0..255]$;
- c) $(0..255)$;
- d) $(-128..127)$.

25. Результатом вычисления $5 \ll 3$ будет:

- a) 40;
- b) 3;
- c) 0;
- d) 12.

26. Операция, позволяющая определить количество байт, которые занимает переменная t :

- a) `(double) t`;
- b) `sizeof(t)`;
- c) `sizeof t`.

27. Позволяет преобразовать переменную t типа `int` к вещественному типу:

- a) `(double) t`;
- b) `(float) t`;

c) (longint) t.

28. После выполнения фрагмента

```
int y=10;
```

```
(y > 15) ? (y *= 10) : (y /= 7);
```

переменная y примет значение:

a) 1;

b) 1.42;

c) 100.

29. В результате работы программы

```
#include <iostream.h>
```

```
#include <conio.h>
```

```
void main (void)
```

```
{  
    short n;  
    n=32770;  
    cout<<"n="<<n;  
    getch();  
}
```

на экран будет выведено:

a) 32770;

b) -32766;

c) 2;

d) выведет сообщение об ошибке.

30. Запишите результат работы программы:

```
#include<stdio.h>
```

```
main(void)
```

```
{ int x=5;  
  printf("x=%d \n",++x);}
```

a) 5;

b) 4;

c) 6;

d) 7.

31. Результат выполнения следующего фрагмента кода: `cout<< 22 / 5 * 3;`

a) 1.47;

b) 1;

c) 12;

d) 13.2.

32. Ключевое слово, указывающее, что целая переменная не может принимать отрицательные значения:

a) unsigned;

b) нет такого зарезервированного слова;

c) positive;

d) long.

ДЕЗ. Структурное программирование – 36

33. Если условие условного оператора ложное, то:

a) выполняется следующий оператор, сразу после оператора if;

b) программа завершает работу;

c) выполняется тело условного оператора.

34. Логическим оператором И является:

a) &&;

b) |;

c) &;

d) ||.

35. Результат выполнения следующего фрагмента кода: `!((1 || 0) && 0)`

a) 0;

b) 1;

c) результат не может быть заранее определен.

36. После выполнения фрагмента кода

```
inta = 1, b = 2; if (a == b); cout<<a<<" = " <<b<<endl;
```

на экране появится:

- a) синтаксическая ошибка;
- b) $a = b$;

- c) вывод на экран не выполнится;
- d) $1 = 2$.

37. Логическая операция с большим приоритетом выполнения:

- a) $\parallel$;
- b) $\&\&$;

- c) $!$.

38. В блок-схеме для записи проверки условия используется фигура:

- a) эллипс;
- b) прямоугольник;

- c) параллелограмм;
- d) ромб.

39. Условный оператор относится к ___ алгоритму:

- a) к линейному;
- b) к циклическому;

- c) к разветвляющемуся;
- d) к основному.

40. Оператор ifelse позволяет определить действие:

- a) для истинного и ложного условий;
- b) только для истинного условия;

- c) только для ложного условия.

41. Синтаксическая ошибка допущена в условном операторе:

- a) `if (B == 0) cout<<"Деление на нуль невозможно";`
- b) `if (a > b) max = a; else max = b;`
- c) `if (a>b) && (b>0) c=a+b;`
- d) `if (a < b) min = a else min = b.`

42. Программа передаёт управление в случае, если значение переменной или выражения оператора switch не совпадает ни с одним константным выражением, зарезервированному слову:

- a) other;
- b) all;

- c) void;
- d) default.

43. Общий формат оператора множественного выбора switch:

- a) switch (N)

```
{
  case M1:    S1;
              break;
  case M2:    S2;
              break;
  .....
  default:   S;
}
```

- b) switch (N)

```
{
  case M1:    S1;
              break;
  case M2:    S2;
              break;
  .....
  else:      S;
}
```

- c) switch (N)

```
{
  case M1:    S1;
              break;
  case M2:    S2;
              break;
```

```

.....
continue: S;
}

```

44. В результате выполнения следующей программы

```

void main()
{
int a, b, c, i;
b=9; c=4;
i=b/c;
switch (i)
{ case 1: a=b+c;
  case 2: { a=b-c; b=b+2; a=a+b-c;}
  case 3: a=2*(b+c);
  case 4: a=b*c;}
cout<<a;
}

```

значение переменной a равно:

- a) 13; c) 26;
b) 12; d) 36.

45. График функции, соответствующий фрагменту программы

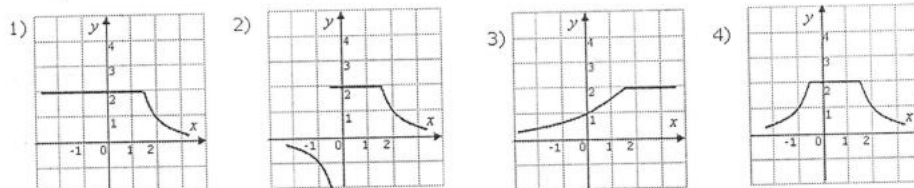
```

if (x<-0.5) y=1/abs(x);
else if (x<1.5) y=2;
else y=1/(x-1);

```

изображен на рисунке:

- a) 1;
b) 2;
c) 3;
d) 4.



46. Определите значение переменной c после выполнения следующего фрагмента программы

```

a = 30;
b = 6;
a = a / 2 * b;
if (a>b)
    c = a - 4 * b;
else
    c = a + 4 * b;

```

- a) 6; c) 26;
b) 66; d) 54.

47. В результате работы фрагмента программы

```

k=2;
d=5;
if ((k+d)<3)
{ m=d-k;
cout<<"m="<<m;
}
else
{ m=d+k;

```

```
cout<<"m"<<m; }
```

на экране монитора появится число:

a) 7;

b) 3;

c) 2;

d) 5.

48. Укажите значение переменной y

```
x=3; n=2
```

```
switch (n)
```

```
{ case 1: y=x*x; break;
```

```
case 2: y=x*x*x; break;
```

```
case 3: y=x+3; break;
```

```
}
```

a) 4;

b) 27;

c) 8.

49. Укажите бесконечные циклы:

a) for (i=1; i++;) printf("цикл\n");

b) for (i=0; i>=1; i++;) printf("цикл\n");

c) for (i=10; i>6; i++;) printf("цикл\n");

d) for (i=5; i>15; i++;) printf("цикл\n").

50. В результате выполнения следующей программы:

```
void main()
```

```
{
```

```
int n,x;
```

```
    n=1;
```

```
    while (n<=5) {
```

```
        x=n*n;
```

```
    cout<<" "<<x;
```

```
        n++;
```

```
    }
```

```
}
```

на экран будет выведено:

a) 1 4 9 16 25;

b) 1 2 3 4 5;

c) 5 4 3 2 1;

d) 25 16 9 4 1.

51. Найдите ошибочно записанные операторы:

a) while (true) i=i+1;

b) j=56;

i=1;

```
while (i<j div 7) i=i+1;
```

c) j=5; while (j>0) j=j+1;

d) j=-5; while (j<=0) j=j+1;

a) a;

b) a,c;

c) c;

d) a,d.

52. Операторы в теле цикла while (x< 100) будут выполняться до тех пор, пока:

a) x строго меньше 100;

b) x меньше или равен 100;

c) x равен 100;

d) x больше 100.

53. Для цикла while верно высказывание:

a) тело цикла в любом случае выполнится хотя бы один раз;

b) цикл с постусловием;

c) цикл с предусловием.

54. Укажите соответствия:

- a) for
- b) while
- c) dowhile

- 1) цикл с предусловием;
- 2) цикл с постусловием;
- 3) цикл со счетчиком.

55. Хотя бы один раз обязательно выполнится:

- a) цикл с предусловием;
- b) цикл с постусловием;

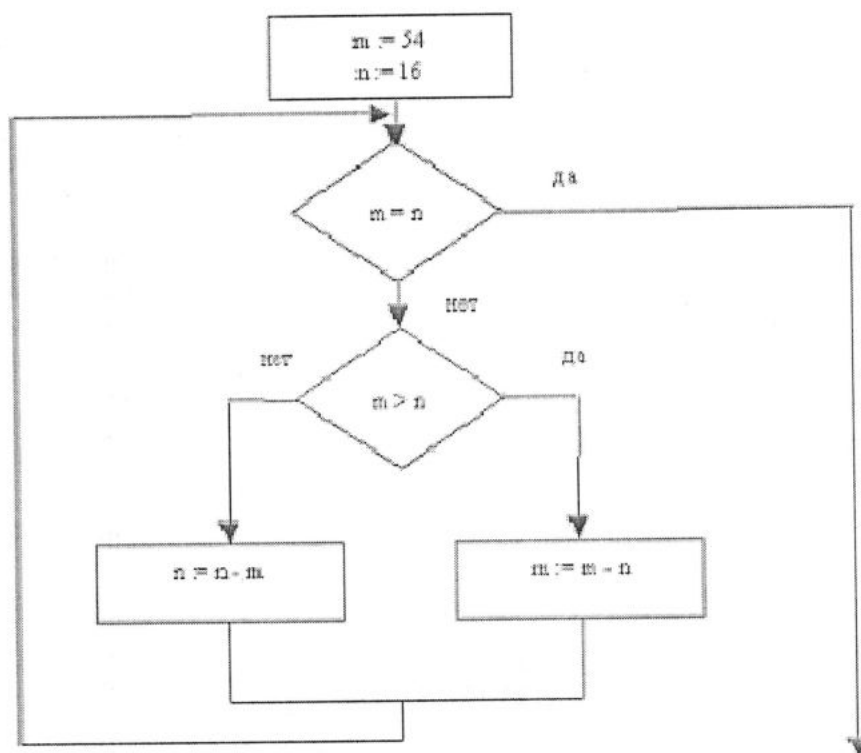
- c) цикл со счетчиком.

56. Не является циклом в C++:

- a) do while;
- b) repeat until;

- c) for;
- d) while.

57. Значение переменной m после выполнения фрагмента алгоритма



будет равно ____.

58. Значение переменной S после выполнения операторов

```

i:=0; S:=0;
while (i<3)
{
i:=i++;
S:=S+i*i;
}
  
```

будет равно ____.

59. Дан фрагмент for ($i=1, s=0; i \leq 3; i++$). Один раз до цикла выполнится:

- a) $i++$;
- b) $i \leq 3$;
- c) $i=1$;
- d) $s=0$.

60. Дан фрагмент for ($i=1, s=0; i \leq 5; i+=2$) $s+=i$;

Укажите значение переменной s , после завершения цикла:

- a) 15;
- b) 4;
- c) 9.

61. В результате выполнения программы

```
void main()
```

```
{
```

```
int n, s;
```

```
n = 30;
```

```
s = 25;
```

```
while (s < 425)
```

```
{
```

```
    s = s + 40;
```

```
    n = n + 10;
```

```
}
```

```
printf("%d", n);
```

```
}
```

будет напечатано:

a) 425;

c) 130;

b) 120;

d) 440.

62. Дан фрагмент `for (i=1, s=0; i<=5 ; i+=2) s+=i;` На каждом шаге цикла будут выполняться:

a) `i+=2;`

b) `i<=5;`

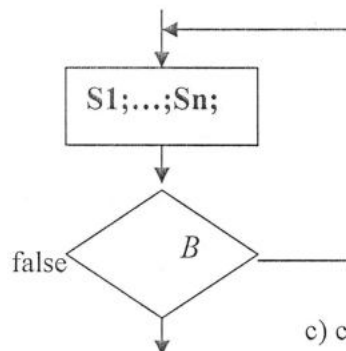
c) `i=1;`

d) `s+=i;`

e) `s=0.`

63. Данная блок-схема

true



соответствует циклу:

a) со счетчиком;

b) с предусловием;

c) с постусловием.

64. Если говорят, что программа "зациклилась", то это значит, что тело цикла:

a) выполняется бесконечно долго;

b) не выполняется;

c) выполняется два раза;

d) выполняется только один раз.

65. Назначение циклической структуры:

a) повторение идущих подряд одинаковых команд некоторое число раз;

b) проверка условия в тексте;

c) печать текста;

d) повторение одной команды не более 10 раз.

66. Прервать выполнение операторов `do`, `for`, `while` и `switch` можно оператором:

a) `break;`

b) `continue;`

c) `exit.`

67. Оператор `Continue` выполняет:

a) переход к началу новой итерации цикла;

b) переход к оператору, следующему за циклом;

c) выход из цикла.

68. Прервать программу позволяет оператор:

a) `break;`

b) `continue;`

c) `exit.`

ДЕ 4. Модульное программирование – 12

69. Подпрограмма – это:

- а) повторяющаяся группа операторов, оформленная в виде самостоятельной программной единицы. Она записывается однократно, а в соответствующих местах программы обеспечивается лишь обращением к ней по имени;
- б) независимая от основной программы группа операторов, оформленная в виде самостоятельной программной единицы. Она записывается однократно, обращение к ней из основной программы не происходит;
- в) повторяющаяся группа операторов, оформленная в виде самостоятельной программной единицы и записанная в отдельный файл.

70. Оператор, возвращающий значение функции:

- а) void;
- б) function;
- в) return;
- г) default.

71. Укажите правильный вызов функции, если функция была объявлена ранее.

- а) funct();
- б) funct x, y;
- в) int funct();
- г) funct.

72. Укажите тип возвращаемого значения функции int func (char x, float v, double t);

- а) float;
- б) double;
- в) int;
- г) char.

73. Выберите правильное (полное) определение функции.

- а) void funct(int)
{
cout<< "Hello"
}
б) void funct(x)
{
cout<< "Hello"
}
- в) int funct();
- г) int funct(int x)
{
return x = x + 1;
}

74. Выполнение программы на языке Си начинается с функции:

- а) main;
- б) void;
- в) default;
- г) include.

75. Максимальное число аргументов, которое можно передавать функции:

- а) 8;
- б) 256;
- в) 512;
- г) 1024.

76. После выполнения программы

```
#include <iostream.h>
#include <conio.h>
double Max(double a, double b)
{
    if (a > b)
        return a;
    else
        return b;
}
void main (void)
{
    double x, y, max;
    x = -5.126;
    y = -5.125;
```

```
cout<<"max="<<Max(x,y)<<endl;
getch();
}
```

на экран будет выведено:

- a) -5.126;
- b) -5.125;

c) 0;

77. Установите соответствия:

- a) Аргумент
- b) Параметр

- 1) значение, передаваемое функции при вызове;
- 2) определяемая функцией переменная, которая принимает передаваемый функции аргумент.

78. Локальными называются переменные:

- a) которые объявлены в описании основной части, и их могут использовать любые функции данной программы;
- b) описанные внутри подпрограммы и используемые только внутри данной подпрограммы;
- c) описанные внутри подпрограммы и используемые как внутри данной подпрограммы, так и в основной программе.

79. Возвращаемое функцией по умолчанию значение принимает тип:

- a) void;
- b) int;
- c) float;
- d) bool.

80. В описании функции, которая не возвращает никаких результатов, используется тип:

- a) void;
- b) int;
- c) float;
- d) bool.

При тестировании все верные ответы берутся за 100%, тогда оценка выставляется в соответствии с таблицей:

Процент выполнения задания	Оценка
85% и более	5 (отлично)
70-84%	4 (хорошо)
50-69%	3 (удовлетворительно)
менее 50%	2 (неудовлетворительно)

Бланк ответов к тестовым заданиям

ДЕ1	
1	d
2	b
3	a
4	b
5	b
6	b
7	b
8	b
9	a
10	b,d
11	a
12	a
13	a, b, c

14	a,b
----	-----

ДЕ2	
15	a
16	a,b,c
17	b
18	a
19	b
20	a2, b4, c1, d3
21	c
22	a
23	b
24	b
25	a

26	b
27	a,b
28	a
29	b
30	c
31	c
32	a

72	c
73	d
74	a
75	b
76	b
77	a1, b2
78	b
79	b
80	a

ДЕ3	
33	a
34	a
35	b
36	a
37	c
38	d
39	c
40	a
41	d
42	d
43	a
44	b
45	4
46	b
47	a
48	b
49	b
50	a
51	b
52	a
53	c
54	a3, b1, c2
55	b
56	b
57	4
58	14
59	c,d
60	c
61	c
62	d
63	c
64	a
65	a
66	a
67	a
68	c

ДЕ4	
69	a
70	c
71	a

«УТВЕРЖДАЮ»

зав. кафедрой информатики и
программной инженерии,
доцент *Л.А. Тягульская*
« 21 » 09 2022 г.

**Тестовые задания для промежуточного контроля
по итогам освоения дисциплины, а также для контроля самостоятельной работы студента
(2 семестр)**

Указания: Напишите Вашу фамилию, номер группы и дату. Для ответа на вопрос с выбором варианта ответа достаточно написать номер вопроса и рядом букву, обозначающую правильный вариант из предложенных в тексте ответов на вопрос. Если Вы считаете правильными несколько вариантов ответов, то запишите через запятую соответствующие литеры букв. Ответ считается верным, если отмечены все правильные варианты ответов.

Время выполнения – 30 минут.

ДЕ 4. Модульное программирование – 9

1. Рекурсия – это:
 - a) вызов нескольких функций с одного адреса в памяти компьютера;
 - b) обращение функции к собственному адресу в памяти компьютера;
 - c) управление параметрами функции из самой себя;
 - d) вызов функции самой себя для выполнения определенных действий.
2. Переменные, которые принимают конкретные значения только во время выполнения подпрограммы, называются:
 - a) конкретными;
 - b) нормальными;
 - c) фактическими;
 - d) формальными.
3. Если подпрограмма вычисляет единственный результат, то ее лучше оформить в виде:
 - a) функции;
 - b) процедуры;
 - c) программы;
 - d) модуля.
4. Переменные, объявленные в основной программе, будут:
 - a) локальными для всех подпрограмм;
 - b) локальными только для первой подпрограммы;
 - c) глобальными для всех подпрограмм;
 - d) глобальными только для первой подпрограммы.
5. Параметры, являющиеся средством передачи данных в одном направлении, – это:
 - a) параметры-переменные;
 - b) параметры-значения;
 - c) формальные параметры;
 - d) фактические параметры.
6. Виды рекурсий:
 - a) непосредственная;
 - b) косвенная;
 - c) обратная;
 - d) двухсторонняя.
7. Оператор присваивания обязательно должен присутствовать в теле:

- a) функции;
- b) процедуры;
- c) программы;
- d) модуля.

8. Укажите разницу между фактическими и формальными параметрами функций:

- a) формальные параметры могут использоваться только вне тела функции, а фактические – используются как вне функции, так и внутри нее;
- b) нет разницы, это одни и те же параметры;
- c) и формальные, и фактические параметры используются вне тела функции;
- d) формальные параметры определены в теле функции, а фактические – значение, с которыми функция вызывается.

9. Укажите, какие спецификаторы доступа не существуют:

- a) public;
- b) protected;
- c) private;
- d) open.

ДЕ 5. Составные типы данных – 13

1. Укажите правильный вариант объявления указателя в C.

- a) int x;
- b) int & x;
- c) int * x;
- d) ptr x;

2. Выберите составные части определения массива:

- a) совокупность переменных одного типа;
- b) переменные хранятся в последовательно расположенных ячейках оперативной памяти;
- c) последовательность переменных, которые имеют одинаковое имя;
- d) благодаря нумерации переменные располагаются только в строку.

3. Укажите, что может быть индексом элемента массива

- a) символ;
- b) дробное число;
- c) целое положительное число;
- d) математическое выражение, результат вычисления которого является целое положительное число.

4. Выберите верные утверждения, имеющие отношение к имени массива:

- a) название массива – ключевое слово array и любая латинская буква;
- b) название массива является указателем на его первый элемент;
- c) название массива является указателем на его элемент с индексом [1];
- d) имя массива создается по правилам создания идентификатора.

5. Выберите верные утверждения, имеющие отношение к размеру массива:

- a) размером могут быть переменные целого типа;
- b) размером могут быть переменные вещественного типа;
- c) размер – это индекс последнего элемента массива.

6. Укажите описания массивов, которые нельзя использовать ни в какой части программы:

- a) Int a [3];
- b) Int a ['f'];
- c) Int a [];
- d) Int a [3] = {1}.

7. В какой из следующих строк, записанных на языке C, выполняется обращение к седьмому элементу массива, с учетом того, что размер массива равен 10?

- a) mas (7);
- b) mas [6];
- c) mas 8;

- d) mas [7];
8. Пусть `int a [5] = {10, 11, 12, 13, 14}`. Значение элемента `a [3]` равно:
- a) 3;
 - b) 12;
 - c) 13;
 - d) 14.
9. Строки в C представляются как массивы элементов типа `char`, заканчивающиеся символом:
- a) `'0'`;
 - b) `'\0'`;
 - c) `»`;
 - d) `'.'`.
10. Тип данных, определяемый служебным словом `struct`, – это:
- a) массив данных с различной структурой;
 - b) тип функций, которые могут иметь различную структуру параметров;
 - c) тип данных, которые могут менять свою структуру;
 - d) составной объект, к которому могут входить элементы различных типов.
11. Элементы, которые может содержать структура:
- a) только поля;
 - b) только функции;
 - c) поля и функции.
12. Типы доступа, возможные для элементов структур:
- a) любые;
 - b) только закрытые;
 - c) только открытые;
 - d) только защищенные.
13. Средствами языка C потоковое чтение из двоичного (бинарного) файла (название потока `a_file`) выполняется:
- a) `a_file >> z;`
 - b) `a_file.in (z);`
 - c) `z = a_file.read;`
 - d) `a_file.read ((char *) & z, sizeof z);`

ДЕ 6. Стандартные библиотеки C++ – 8

1. Модули предназначены для:
- a) хранения данных базы;
 - b) ввода данных базы и их просмотра;
 - c) автоматического выполнения группы команд;
 - d) выполнения сложных программных действий.
2. Функция `GetColor`:
- a) устанавливает текущий цвет;
 - b) возвращает код текущего цвета;
 - c) устанавливает цвет фона;
 - d) возвращает значение, представляющее собой код используемого цвета фона.
3. Процедура отображения прямой линии между двумя точками, координаты которых заданы в качестве параметров процедуры:
- a) `Line`;
 - b) `LineTo`;
 - c) `SetLineStyle`;
 - d) `GetLineStyle`.
4. Процедура `Rectangle`:

- a) рисует контур многоугольника, используя текущий цвет и стиль линии;
 - b) отображает прямоугольник с использованием текущего стиля и цвета линии;
 - c) отображает трехмерный параллелепипед с заполненной передней гранью;
 - d) заполненный многоугольник.
5. Библиотеки подключаются к программе с помощью:
- a) `uses;`
 - b) `unit;`
 - c) `{;`
 - d) `include.`
6. Для работы с функциями управления экраном (очистка, задержка) используется библиотека:
- a) `stdlib;`
 - b) `iostream;`
 - c) `conio;`
 - d) `math.`
7. Процедура `Bar` отображает:
- a) прямоугольник без использования шаблона и цвета;
 - b) прямоугольник с использованием текущего шаблона и цвета;
 - c) трехмерный параллелепипед с заполненной передней гранью;
 - d) заполненный многоугольник.
8. Работа с текстом возможна в _____ режиме(ах):
- a) графическом;
 - b) текстовом;
 - c) и графическом, и текстовом;
 - d) ни в графическом, ни в текстовом.

ДЕ 7. Введение в объектно-ориентированное программирование – 6

1. Принцип полиморфизма в С реализуется с помощью:
- a) наличия множественного наследования;
 - b) наличия виртуальных методов;
 - c) использования виртуального наследования;
 - d) наличия абстрактных классов.
2. Принцип объектно-ориентированного программирования, который заключается в объединении атрибутов и методов объекта с целью обеспечения сохранности данных, называется:
- a) наследование;
 - b) сочетание;
 - c) инициализация;
 - d) инкапсуляция.
3. Выберите верное утверждение о деструкторе класса в С.
- a) деструктор принимает в качестве параметра адрес того объекта, который нужно уничтожить;
 - b) деструктор принимает в качестве параметра указатель `this`;
 - c) деструктор не содержит параметров;
 - d) деструктор принимает в качестве параметра тот объект, который нужно уничтожить.
4. Конструкторы класса предназначены для:
- a) конструирования класса;
 - b) инициализации объектов класса;
 - c) выделения памяти объектам класса;
 - d) объявления объектов класса.
5. Дан фрагмент кода, содержащий вызов метода класса:
- `Pro p1 (2,5)`

p1.Show ();

Определить прототип функции-элемента класса:

- a) Void Show (int a, int b);
- b) Void pro :: Show (int a, int b);
- c) Void Show (Pro &);
- d) Void Show ();

6. Количество полей и методов в объявлении класса:

```
class Tovar
```

```
{Char * nazva; int price;
```

```
void Show ();
```

```
Tovar ();
```

```
Tovar (char *, int);
```

```
};
```

- a) 2 поля, 1 способ;
- b) 2 поля, 2 метода;
- c) 2 поля, 3 метода;
- d) 3 поля, 2 метода;
- e) 5 полей, метод нет.

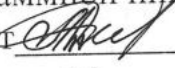
При тестировании все верные ответы берутся за 100%, тогда оценка выставляется в соответствии с таблицей:

Процент выполнения задания	Оценка
85% и более	5 (отлично)
70-84%	4 (хорошо)
50-69%	3 (удовлетворительно)
менее 50%	2 (неудовлетворительно)

Бланк ответов к тестовым заданиям

Номер задания	ДЕ 4	ДЕ 5	ДЕ 6	ДЕ 7
1	d	c	d	b
2	c	a, b, c	b	d
3	a	c, d	a	c
4	c	b, d	b	c
5	b	a	d	d
6	b	b	d	c
7	a	b	b	—
8	d	c	c	—
9	d	b	—	—
10	—	d	—	—
11	—	a	—	—
12	—	c	—	—
13	—	d	—	—

«УТВЕРЖДАЮ»

зав. кафедрой информатики и
программной инженерии,
доцент  Л.А. Тягульская
« 21 » 09 2023 г.

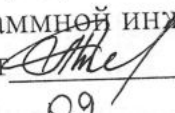
**Вопросы к экзамену
по дисциплине «Программирование»
для студентов I курса
направления «Программная инженерия»
профиля «Разработка программно-информационных систем»,
1 семестр**

1. Основные понятия программирования.
2. Языки программирования: понятие, структура, классификация, основные элементы.
3. Язык высокого уровня C. Структура программы.
4. Системы программирования: понятие, структура, виды.
5. Трансляторы: понятие, виды.
6. Компилятор. Компиляция: виды, этапы работы компилятора.
7. Интерпретатор. Интерпретация: виды, этапы работы интерпретатора.
8. Принцип построения программ на языке C (модульная организация).
9. Назначение стандартных заголовочных файлов. Библиотека стандартных функций.
10. Организация ввода данных. Входной поток. Форматированный вывод. Спецификаторы формата.
11. Управляющие символы. Отличия в применении операторов ввода *cin* и *scanf*, операторов вывода *cout* и *printf*.
12. Понятие типа данных. Классификация типов данных. Модификаторы типа.
13. Целые типы данных: *int*, *short*, *long*, *char*. Кодовый формат. Тип целых констант.
14. Представление целых чисел на компьютере, специальные коды.
15. Арифметические и логические операции для целых операндов.
16. Вещественные типы данных: *float*, *double*. Кодовый формат. Тип констант с плавающей точкой.
17. Представление вещественных чисел на компьютере. Научный формат.
18. Арифметические операции для вещественных операндов.
19. Математические функции стандартной библиотеки C из объектных модулей и библиотек.
20. Псевдослучайные числа.
21. Преобразование типа данных: понятие, виды. Правила преобразования операндов в процессе вычислений.
22. Оператор-выражение. Простой и составной операторы. Оператор присваивания.
23. Знаковые операторы. Особенности использования постфиксной и префиксной формы операторов инкремента и декремента.
24. Операторы отношения. Представление булевских значений «ложь» и «истина» в C.
25. Побитовые и логические операторы.

26. Создание *Windows*-приложений в среде *Borland C++ Builder*. Менеджер проектов.

Экзаменатор, ст. преподаватель _____  Е.С. Гарбузняк

«УТВЕРЖДАЮ»

зав. кафедрой информатики и
программной инженерии,
доцент  Л.А. Тягульская
« 21 » 09 2023 г.

**Вопросы к экзамену
по дисциплине «Программирование»
для студентов I курса
направления «Программная инженерия»
профиля «Разработка программно-информационных систем»,
2 семестр**

1. Область действия переменных. Локальные и глобальные переменные. Видимость переменных.
2. Время жизни переменной. Модификаторы переменных. Изменяющиеся переменные.
3. Рекурсивные подпрограммы. Понятие о рекурсии. Сравнение рекуррентности и рекурсии.
4. Прямой и обратный ход рекурсии. Точка останова работы рекурсивного алгоритма. Глубина рекурсии.
5. Обеспечение корректной стыковки модулей. Создание библиотек функций.
6. Создание заголовочных файлов.
7. Массивы: понятие, виды, примеры использования. Обращение к элементу массива. Работа с элементами массива.
8. Многомерные массивы. Алгоритмы работы с многомерными массивами.
9. Сортировки массивов: пузырьковая, выбором, вставкой.
10. Указатели. Указатели и массивы. Указатели и функции.
11. Тип данных строки. Функции работы со строками. Таблица *ASCII*.
12. Библиотека *<string.h>*.
13. Структура как тип данных. Поле структуры. Обращение к полям структуры. Размещение структур в памяти. Примеры использования.
14. Типы, определяемые пользователем. Объединения.
15. Типы, определяемые пользователем. Перечисления.
16. Типы, определяемые пользователем. Переименование типов данных.
17. Основные понятия класса. Конструкторы и деструкторы.
18. Классы и структуры. Взаимосвязь и отличия.
19. Классы и объединения. Взаимосвязь и отличия.
20. Тип данных файлы. Работа с файлами. Поточковый ввод/вывод данных.
21. Понятие о динамических данных. Линейные списки. Организация данных в виде стека, очереди, дерева.
22. Обработка исключительных ситуаций.
23. Графика. Структура графической программы. Стандартные функции для работы с графикой.
24. Ведение в объектно-ориентированное программирование. Объекты и классы.
25. Среда разработки *Borland C++ Builder*. Основные компоненты. Свойства. События. Методы.

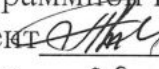
26. Тернарный оператор. Приоритет и ассоциативность операторов.
27. Разветвляющиеся алгоритмы. Условный оператор *if*. Сложные условия.
28. Вложенные *if*-инструкции. Конструкция *if-else-if*.
29. Оператор многонаправленного ветвления *switch*. Использование инструкций *break* и *default*. Вложенные инструкции *switch*.
30. Цикл: понятие, виды, организация.
31. Оператор многократного выполнения *for*, особенности использования. Вложенные операторы цикла.
32. Оператор повторений *while*, особенности использования. Вложенные операторы цикла. Замена *for* на *while* и наоборот.
33. Оператор повторений *do ... while*, особенности использования. Вложенные операторы цикла.
34. Отсутствие элементов в определении цикла. Бесконечные циклы. Циклы временной задержки.
35. Инструкции управления циклами: оператор перехода *continue*, оператор завершения цикла *break*.
36. Инструкция *goto*.
37. Функции: общий формат, аргументы и параметры. Передача аргументов по значению и по ссылке.
38. Прототипы функций. Преобразование аргументов в точке вызова. Инструкция *return*.

Экзаменатор, ст. преподаватель _____



Е.С. Гарбузняк

«УТВЕРЖДАЮ»

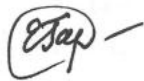
зав. кафедрой информатики и
программной инженерии,
доцент  Л.А. Тягульская
« 21 » 09 2023 г.

**Примерные темы курсовых работ
по дисциплине «Программирование»
для студентов I курса
направления «Программная инженерия»
профиля «Разработка программно-информационных систем»,
2 семестр**

1. Разработка программы автоматизации формирования модели бюджета.
2. Программный учёт материально-технических средств на примере предприятия.
3. Автоматизация поставки продуктов в гостиничный комплекс.
4. Разработка программы мониторинга продукции, представленной в интернет-магазинах.
5. Разработка программы управления проектами.
6. Разработка сервиса для тестирования знаний.
7. Моделирование симулятора планетарной системы.
8. Разработка корпоративного мессенджера с геопривязкой.
9. Разработка системы планирования пассажирских автотранспортных перевозок.
10. Разработка системы резервирования билетов.
11. Разработка программы автоматизации учёта занятости аудиторий в вузе.
12. Программирование учёта товарно-материальных ценностей в розничном магазине.
13. Программа рисования графического орнамента.
14. Создание анимации объектов.
15. Личный кабинет студента.
16. Подсистема размещения и хранения научных проектов кафедры.
17. Автоматизация отчетных ведомостей.
18. Электронный журнал текущей успеваемости и посещаемости студентов.
19. Реализация функциональных возможностей для абитуриентов в рамках сайта кафедры.
20. Реализация новостной ленты с возможностью привязки к социальным сетям.
21. Разработка информационно-поисковой системы.
22. Разработка игрового приложения «Кроссворд».
23. Программа тестирования знаний студентов по дисциплине.
24. Программа шифрования и дешифрования текстовых файлов.
25. Программа, моделирующая тренажер по устному счету.
26. Игровое приложение «Ребус».
27. Программа для построения графиков математических функций.
28. Программа, реализующая функции сопровождения базы данных библиотеки.
29. Программа для игры в «Морской бой».
30. Программа, автоматизирующая построение фигур на плоскости.
31. Программа, моделирующая строковый калькулятор.

32. Разработка простейшего почтового клиента.
33. Программа, моделирующая работу стрелочных часов с кукушкой.
34. Программа автоматического написания характеристик.
35. Разработка графического редактора для геометрического отображения графов.
36. Разработка обучающе-контролирующей программы по программированию.
37. Реализация алгоритма Фон Неймана.
38. Демонстрация возможностей работы с графическим файлом.

Ст. преподаватель



Е.С. Гарбузник