Министерство образования Иркутской области
Государственное бюджетное профессиональное
образовательное учреждение Иркутской области
«Иркутский авиационный техникум»

Методические указания по выполнению самостоятельной работы по дисциплине

ОП.04 Основы алгоритмизации и программирования

специальности

09.02.07 Информационные системы и программирование

Иркутск, 2024

РАССМОТРЕНЫ

Председатель ЦК

_____ / /

УТВЕРЖДАЮ
Зам. директора по УР

_____ Е.А. Коробкова

№	Разработчик ФИО
1	Филимонова Ольга Николаевна

Пояснительная записка

Дисциплина ОП.04 Основы алгоритмизации и программирования входит в Общепрофессиональный цикл. Самостоятельная работа является одним из видов учебно работы обучающегося без взаимодействия с преподавателем.

Основные цели самостоятельной работы:

- систематизация и закрепление теоретических знаний и практических умений обучающихся;
- углубление и расширение теоретических знаний, формирование умений использовать справочную документацию и дополнительную литературу;
- развитие познавательных способностей и активности обучающихся, творческой инициативы, самостоятельности, ответственности и организованности; формирование самостоятельного мышления;
- развитие исследовательских умений.

Методические рекомендации помогут обучающимся целенаправленно изучать материал по теме, определять свой уровень знаний и умений при выполнении самостоятельной работы. На самостоятельную работу в курсе изучения дисциплины Основы алгоритмизации и программирования отводится 4 часа. Вид самостоятельной работы - аудиторная.

Рекомендации для обучающихся по выработке навыков самостоятельной работы:

- Слушать, записывать и запоминать лекцию.
- Внимательно читать план выполнения работы.
- Выбрать свой уровень подготовки задания
- Обращать внимание на рекомендуемую литературу. Из перечня литературы выбирать ту, которая наиболее полно раскрывает вопрос задания
- Учиться кратко излагать свои мысли
- Использовать общие правила написания конспекта,
- Обращать внимание на достижение основной цели работы.

Тематический план

Раздел Тема	Тема занятия	Название работы	Количество часов
Раздел 1. Программирование на языке С Тема 1. Основы алгоритмизации	Способы обработки элементов массива	Способы обработки элементов массива	2
Раздел 2. Основы объектно-ориентированного программирования Тема 1. Основные принципы объектно-ориентированного программирования	Поиск ошибок в коде.	Поиск ошибок в коде.	2

Самостоятельная работа №1

Название работы: Способы обработки элементов массива.
Цель работы: Выявление умения работать с учебным текстом (изучение нового материала)..
Уровень СРС: эвристическая.
Форма контроля: Решение задач.
Количество часов на выполнение: 2 часа.
Задание:
Ознакомьтесь с материалом по ссылке: Основные виды сортировок и примеры их реализации (yandex.ru).
Разберите примеры, составьте коды программы для приведенных способов сортировки на изучаемом языке программирования (С++).
Решите задачи данными способами:

1. Если в массиве есть более одного элемента кратного 13, то отсортировать массив по возрастанию, иначе – по убыванию.
2. Если сумма отрицательных элементов по модулю превышает сумму положительных, то отсортировать массив по возрастанию, иначе – по убыванию.
3. Если в массиве есть нулевые элементы, то отсортировать массив по возрастанию, иначе – по убыванию.

Критерии оценки:
оценка «3» - Код программы написан для 3х примеров, решена одна задача;
оценка «4» - Код программы приведен для 3х примеров, решено две задачи;
оценка «5» - Код программы приведен для 3х примеров, решено три задачи.

Самостоятельная работа №2

Название работы: Поиск ошибок в коде..
Цель работы: Закрепление изученного материала..
Уровень СРС: эвристическая.
Форма контроля: Решение задач..
Количество часов на выполнение: 2 часа.
Задание:
Ниже приведен текст программы на изучаемом языке программирования (С++).
Рассмотрите данный пример, опишите в чем заключается суть программы.
Выполните проверку, отладку кода программы.
Готовый результат и описание представьте в текстовом документе со скриншотом выполненной программы, напишите комментарии к каждой строке кода.

```
include <string>
class students {
public
```

```
void set_name(string student_name)
{
name = student_name;
}
String get_name()
{
return name;
}
void set_last_name(string student_Last_name)
last_name = student_last_namne
string get_last_nane()
return last_name
void set_scores(int student_scores[])
{
For (int i=0; i<5;i++) {
Scores[i]=student_scores[i];
}
}
void set_average_ball(float ball)
average_ball = ball;
Float get_average_ball()
return average_ball;
private:
int scores [5];
float average_ball;
string name;
string last_name;
};
int main()
{
Students student;
string name, last_name;
cout << “Введите имя ”;
getline(cin, name);
cout << “Введите фамилию ";
getline(cin, Last_name) ;
student set_name(name);
student. set_last_name(1ast_name);
int scores[5]
int sum=0;
for(int i=0; i<5; ++i){
```

```
cout<<”    “<<i+1<<”: “;
cin>>scores[i];
sum+=scores[i];
}
Student.set_scores(scores);
Float average_ball=sun/5.0;
Student. set_average_ball(average_ball);
Cout<<”Средний балл для “<<students.get_name()<< “ “
<< student. get_last_name<< “ = “
<< Student. get_average_ball()<<endl;
Return 0;
}
```

Критерии оценки:

оценка «3» - Исправлены ошибки в классе Students. Подписаны комментарии. Программа не работает

оценка «4» - Исправлены ошибки в классе Students. Исправлены ошибки в главной функции. Подписаны комментарии. Программа не работает.

оценка «5» - Исправлены ошибки в классе Students. Исправлены ошибки в главной функции. Подписаны комментарии. Программа работает верно.