

**Тесты для выявления остаточных знаний по специальности
«Вычислительные машины, комплексы, системы и сети» на
2010/2011 учебный год**

- (1) [10 баллов]: Истина или Ложь?
 - А. И / Л : Перегрузка операторов в объектно-ориентированных языках программирования позволяет расширить список синтаксических конструкций, используемых языком программирования
 - Б. И / Л : Диаграммы деятельности позволяют представить общие требования к функциональному поведению проектируемой системы
 - В. И / Л : Перегрузка операторов реализована во всех современных языках программирования
 - Г. И / Л : Стражи включения появились в языке программирования C++, как результат его развития по отношению к языку C
- (2) [10 баллов]: Приведите пример интерфейса класса, реализующего интерфейс списка строк. Поясните свой вариант интерфейса с точки зрения адекватности, полноты, ортогональности и простоты.
- (3) [10 баллов]: Поясните особенности применения спецификатора `static` для функций и переменных в классах. Приведите пример объявления и определения статической переменной класса.
- (4) [10 баллов]: Приведите пример объявления класса без конструктора по умолчанию на языке C++. Реализуйте алгоритм инициализации массива из 100 элементов, объявленного класса.
- (5) [10 баллов]: Детально опишите образец проектирования «одиночка», приведите пример использования такого образца.
- (6) [10 баллов]: Опишите образец организации «инспекция программ по Фагану». Приведите пример варианта несоответствия, которое можно выявить между реализацией и тестированием.
- (7) [5 баллов]: Поясните какой из следующих видов тестирования можно или нельзя отнести к видам основанным на стратегии исследования чёрного ящика: модульное, интеграционное и системное тестирование.
- (8) [5 баллов]: Приведите и опишите семь различных видов тестирования.
- (9) [5 баллов]: Приведите список диаграмм UML, с которыми вы знакомы, и дайте каждому из них краткое описание.
- (10) [15 баллов]: Отладка. Найдите и исправьте синтаксические и семантические ошибки в следующем примере. Поясните каждую найденную вами ошибку. Уделите отдельное внимание проблеме утечки памяти — найдите и укажите на проблему связанную с тем, что при удалении объектов память не будет освобождена.

```
class Driver
{
public:
    virtual bool Open(std::string) = 0;
    virtual void Close() = 0;

    virtual std::string Read() = 0;
    virtual void Write(std::string) = 0;
};

class Driver0: Driver
{
public:
    Driver0()
    {
        data = new char[100];
    }
    ~Driver0()
```

```
        {
            delete[] data;
        }

        bool Open(std::string);
        void Close();
protected:
        int *data;
};

class Driver1: Driver
{
public:
    Driver1()
    {
        data = new char[100];
    }
    ~Driver1()
    {
        delete[] data;
    }

    bool Open(std::string);
    void Close();
protected:
    int *data;
};

void ReadFromAll(Driver** drvs)
{
    while (!(*drvs++))
    {
        (*drvs)->Open();
        cout << (*drvs)->Read();
    }
}

void Work()
{
    Driver *drivers[2];
    Driver **drvs = drivers;

    drivers[0] = new Driver0;
    drivers[1] = new Driver1;
    drivers[2] = 0;

    ReadFromAll(drivers);
    while (!(*drvs++))
    {
        (*drvs)->Close();
        delete *drvs;
    }
}
```