

## Контролно упражнение №2 по СДП

Име ..... Група..... Фак. №.....

**Задача 1.** Какво прави (извежда) тази програма? Обяснете как сте получили отговора.

```
#include <iostream.h>
class A
{public:
    A(int nValue)
    {cout << "A: " << nValue << endl;}
};
class B: public A
{public:
    B(int nValue, double dValue)
    : A(nValue)
    {cout << "B: " << dValue << endl;}
};
class C: public B
{public:
    C(int nValue, double dValue, char chValue)
    : B(nValue, dValue)
    {cout << "C: " << chValue << endl;}
};
int main()
{ C cClass(5, 4.3, 'R');
  return 0; }
```

A: 5  
B: 4.3  
C: R

**Задача 2.** Какво прави (извежда) тази програма? Обяснете как сте получили отговора.

```
#include <iostream.h>
class PoweredDevice
{
public:
    PoweredDevice(int nPower)
    {
        cout << "PoweredDevice: " << nPower <<
endl;
    }
};
class Scanner: virtual public PoweredDevice
{
public:
    Scanner(int nScanner, int nPower)
    : PoweredDevice(nPower)
    { cout << "Scanner: " << nScanner << endl;
    }
};
class Printer: virtual public PoweredDevice
{ public:
    Printer(int nPrinter, int nPower)
    : PoweredDevice(nPower)
    {
        cout << "Printer: " << nPrinter << endl;
    }
};
```

```
};
class Copier: public Scanner, public Printer
{ public:
    Copier(int nScanner, int nPrinter, int nPower)
    : Scanner(nScanner, nPower),
    Printer(nPrinter, nPower),
    PoweredDevice(nPower)
    {
    }
};
int main()
{ Copier cCopier(1, 2, 3);
}
```

PD: 3  
SC: 1  
P: 2

**Задача 3.** Напишете какъв резултат извежда следващата програма:

```
#include <iostream.h>
class Point{
public:
    Point(float=0.0, float=0.0);
    ~Point();
protected:
    float x, y;
};
Point::Point(float a, float b)
{ x = a; y = b;
  cout << "Конструктор на Point: "
  << '[' << x << ", " << y << ']' << endl;
}
Point::~~Point()
{ cout << "Деструктор на Point: "
  << '[' << x << ", " << y << ']' << endl;
}
class Circle:public Point{
public:
    Circle(float r=0.0, float x=0.0, float y=0.0);
    ~Circle();
private:
    float radius;
};
Circle::Circle(float r, float a, float b):Point(a, b)
{ radius = r;
  cout << "Конструктор на Circle: " << radius
  << '[' << a << ", " << b << ']' << endl;
}
Circle::~~Circle ()
{ cout << "Деструктор на Circle: " << radius
  << '[' << x << ", " << y << ']' << endl;
}
void main()
{
    Point p(1.1, 2.2);
    Circle circle1(4.5, 7.2, 2.9);
    Circle circle2(10, 5, 5);
}
```

**Задача 4.** Да се открият, обяснят и коригират грешките в следния фрагмент от дефинирането на клас Date и предефиниращата функция на операцията ++ в постфиксна форма.

```
class Date {
public:
...
Date &operator++(int);
private:
int month; // Месец
int day; // Ден
int year; // Година
void helpIncrement(); //Обслужваща функция за
увеличаване на датата с един ден
};
Date &Date::operator++(int)
{ Date temp = *this;
helpIncrement();
return temp;}
```

**Задача 5.** Какъв ще бъде резултатът от изпълнението на следния програмен фрагмент:

```
struct CN {char character ; CN *next ;} *first, cn ;
void list ( char * word )
{ cn.character = *word ; cn.next = first ;
first = new CN ; *first = cn ;
if(*word) list(word+1) ;
}
void print ( CN *first )
{ if(!first) {cout<<"\n result = " ; return ; }
else {print(first->next) ; cout<<first->character ;
return ; }
}
```

```
void main ()
{ first = NULL ;
list( "droll" ) ;
print(first->next->next) ; }
```

Обяснете как сте получили отговора.

**Задача 6.** Функцията search търси елемент по ключ в свързан списък и го актуализира.

Попълнете липсващите оператори на местата на двете многоточия в следния фрагмент от програма:

```
struct student
{ long nomer; // факултетен номер
float sr_uspeh; // среден успех
student *next;};
// first – указател към началото на свързания
списък
// key – ключ. факултетният номер се
разглежда като ключ
// value – стойност, която ще актуализира
средния успех
```

```
void search(student *first, long key, float value)
{ student *ptr = first;
while (ptr)
if (ptr->nomer == key)
break;
.....
else
ptr = ptr -> next
.....
if(ptr)
ptr->sr_uspeh = value;
else
cout << "\nНяма елемент с указания ключ";}
Обяснете как сте получили отговора.
```

**Задача 7.** Функцията push добавя елемент, а функцията pop изключва елемент от последователен стек. Многоточието да се замени с липсващия оператор. Обяснете как сте получили отговора.

```
#define M 100
struct stack
{ int t; // номер на елемента във върха на стека
float stack_array[M];};
a) void push(stack *s, float x)
{ ... ..
if(s->t > M)
{ cout << "\nПРЕПЪЛВАНЕ\n";
exit(1); }
s->stack_array[s->t - 1] = x;
}
b) float pop(stack *s)
{ if (s->t) == 0 exit(1);
s->t--;
return s->stack_array[s->t];
}
```

**Задача 8.** Функцията enqueue добавя елемент в последователна опашка. Попълнете липсващия оператор на мястото на многоточието в следния фрагмент от програма. Обяснете как сте получили отговора.

```
#define M 100
typedef float ElementType;
struct Queue {
int f; // номер на фиктивен елемент преди
началото на опашката
int r; // номер на елемента в края на опашката
ElementType QueueArray[M];};
void enqueue (struct Queue *q, ElementType x)
{ if ((q->r + 1) % M == q->f)
{ cout << "Препълване!" << endl;
exit(1);}
q->r = (q->r + 1) % M
.....
q->QueueArray[q->r] = x;
}
```