X36PJC – test 2 - 6. 5. 2010 const char name[] = _____

1) Doplňte následující kód do funkční podoby. Vytvořte třídu *String* s metodami *setText()* a *getText()*. Textová data musí být ve třídě uložena dynamicky! (Ve třídě *String* se doporučuje použít funkce *strcpy(char *cil, char *zdroj)* - kopírovaní textoveho řetězce a *strlen(char *text)* - zjištění délky textového řetězce.) Do pole ukazatelů (!) „pole“ vložte ukazatele na dvě dynamicky vytvořené instance objektu *String*. Hlavním úkolem je správně alokovat a dealokovat všechnu použitou paměť.

```
int main(){
        String **pole;
        // alokace pameti pole ukazatelu a jednotlivych objektu String
        pole[0]->setText("Nula");
        pole[1]->setText("Jedna");
        cout << pole[0]->getText() << endl;
        cout << pole[1]->getText() << endl;
        // dealokace všech objektu String a pameti pole ukazatelu
        return 0;
}
```

Řešení:

```
        // alokace pameti pole ukazatelu a jednotlivych objektu String
        pole = (String **) malloc(sizeof(String *) * 10);
        pole[0] = new String;
        pole[1] = new String;

        // dealokace všech objektu String a pameti pole ukazatelu
        delete pole[0];
        delete pole[1];
        free(pole);


class String{
        char *text;
public:
        String(){ text = NULL;}
        void setText(char *_text){
                text = new char[strlen(_text)];
                strcpy(text, _text);  }
        char *getText(){
                return text; }
        ~String(){
                if(text != NULL) delete [] text;}
};
```

(2)

2) Napište dva hlavní účely funkce realloc() a ukázku jejího použití.

- alokace paměti
- změna velikosti alokované paměti:

```
char *pole = (char *) realloc(NULL, sizeof(char) * 10);
pole = realloc(pole, sizeof(char) * 20);
```

(1)

3) Napište třídu Integer, která bude splňovat následující chování:

```
int main() {
        Integer number = 4; //ekvivalent Integer number(4);
        cout << 5 + number + 10 << endl; //vypise 19
        cout << number + 1 + number + 5 << endl; //vypise 14
        return 0;
}
```

Jedno z možných řešení:

```
class Integer {
        int value;
public:
        Integer() { value = 0; }
        Integer(int _value) { value = _value; }
        int operator+(int second) { return value + second;}
        friend int operator+(int first, Integer &second) {  return first + second.value; }
};
```

(3)

4) V následujícím kódu je třeba provést jednu úpravu, aby byl přeložitelný:

```
class Base {
protected:     //public není uplne vhodna oprava, porusi se tim logika objektu
        int count;
protected:
        int ID;
public:
        Base(){ count = 0; }
        Base(int ID){ this->ID = ID; }
};
class Child : public Base {
private:
        int age;
public:
        Child(int _age){ age = _age; }
        void increment(){ count++; } //pristup k privatni polozce predka - chyba
};
```

(1)

5) Jaký text bude po provedení kódu vypsán ve standardním výstupu?

```
class First {
public:
        virtual void print() { cout << "First." << endl; }
};
class Second : public First {
public:
        void print() { cout << "Second." << endl; }
};
class Third : public Second {
public:
        virtual void print() { cout << "Third." << endl; }
};
int main(){
        Second *objekt = (Second *) new Third;
        objekt->print();
        return 0;
}
```

Bude vypsán text „Third“.

(1)

6) Co se vypíše do standardního výstupu po provedení následujícího kódu?

```
int funkce(int number){
        static int retval = 0;
        switch (number){
        case 0:
                retval++;
        case 1:
                retval = retval * retval;
                break;
        case 2:
                retval *= retval;
        case 3:
                retval = 3;
                break;
        default:
                retval++;
        }
        return retval;
}
int main() {
        cout << funkce(2) << endl;    // vrati 3
        cout << funkce(0) << endl;    // vrati 16
        return 0;
}
```

(1)

7) Zkuste napsat, jaký text bude po provedení následujícího kódu v souboru test.txt:

```
int main() {
        ofstream test("test.txt");
        (test<<"Cislo: 0x"<<hex<<52).seekp(0,ios::beg).write("He",2).seekp(7,ios::beg)<<dec<<12;
        test.close();
        return 0;
}
```

V souboru bude: „Heslo 1234“. (1)