Moodle VŠPJ

Objektově orientované programování

Započetí testu	Čtvrtek, 1. červen 2023, 10.00
Stav	Dokončeno
Dokončení testu	Čtvrtek, 1. červen 2023, 10.25
Délka pokusu	25 min.
Známka	**23,25** z možných 30,00 (**78**%)

Navigace testu

1 2 3 4 5 6 7 8 9
10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26 27
28 29 30

Zobrazovat po stránkách
Dokončit prohlídku

Úloha **1**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Je dána definice třídy Data.

```
class Data
{
private:
    int d;

public:
    void set_d(int d) const
    {
        this->d = d;
    }

    int get_d() const
    {
        return d;
    }
};
```

Za jistým účelem se v jazyce C++ u objektového programování používá klíčové slovo *const*. Lze i metodu *set_d* označit klíčovým slovem *const*?

Vyberte jednu z nabízených možností:
○ Pravda
● Nepravda

Úloha **2**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Jakými způsoby se řeší předávání informací o chybě z místa vzniku do místa obsluhy?

☑ a. Pomocí globálních příznaků.
☐ b. Pomocí služeb preprocesoru.
☑ c. Pomocí výjimek.
☑ d. Pomocí návratových hodnot.

Úloha **3**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Pro uvedenou definici třídy:

```
class Data
{
private:
    int d;

public:

    Data(int d)
    {
        this->d = d;
    }

    int get_d() const
    {
        return d;
    }

    int value() const
    {
        return d * 4;
    }

    void calc(int v)
    {
        d = v * 0.25;
    }

private:
    void set_d(int d)
    {
        this->d = d;
    }

};
```

Které prvky třídy patří do *"veřejného rozhraní"*?

● a. Data(int d), get_d(), value(), calc(int v),
○ b. Pouze *Data(int d), get_d(), value()*
○ c. *Data(), protože je implicitně definovaný; Data(int d), get_d(), value(), calc(int v)*
○ d. Pouze metody *get_d()* a *value()*
○ e. Data(), protože je implicitně definovaný; Data(int d), get_d(), value()
○ f. Pouze metody *get_d()* a *value()*, protože tyto metody jsou označeny klíčovým slovem **const**.
○ g. Data(int d), get_d(), value(), set_d(int d)
○ h. Data(), protože je implicitně definovaný; Data(int d), get_d(), value(), calc(int v), set_d(int d)

Úloha **4**
Hotovo
Bodů 0,75 / 1,00
Úloha s vlaječkou

Jaké typy konstruktorů existují? Vyberte smysluplné možnosti.

☐ a. Implicitní konstruktor - pouze bezparametrický
☑ b. Bezparametrický konstruktor.
☑ c. Explicitní konstruktor - bezparametrický a parametrický
☐ d. Explicitní konstruktor - pouze parametrický
☑ e. Parametrický konstruktor.
☐ f. Explicitní konstruktor - pouze bezparametrický
☑ g. Implicitní konstruktor - pouze parametrický
☐ h. Implicitní konstruktor - bezparametrický a parametrický

Úloha **5**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Používá se v jazyce C++ pojem "přetížení funkcí" a "přetížení metod"?

Vyberte jednu z nabízených možností:
● Pravda
○ Nepravda

Úloha **6**
Hotovo
Bodů 0,00 / 1,00
Úloha s vlaječkou

V objektově-orientovaném programování se používají pojmy "třída" a "objekt". Vyberte správné tvrzení:

☐ a. *Třída* popisuje objekt a *objekt* je pak konkrétní příklad třídy.
☐ b. *Třída* je proměnnou *objektu*.
☑ c. *Objekt* je proměnnou *třídy*.
☑ d. Pojmy "třída" a "objekt" se používají v dědičnosti. *Třída* je rodič a *objekt* je potomek.

Úloha **7**
Hotovo
Bodů 0,50 / 1,00
Úloha s vlaječkou

Co platí o metodě konstruktoru? Vyberte správná tvrzení.

☑ a. Může mít parametry a může být definována vícekrát (překrytí metody konstruktoru).
☐ b. Provádí se pouze(!) v okamžiku vytváření objektu.
☐ c. Má návratový typ void.
☐ d. Provádí se při rušení objektu.
☐ e. Metoda je pouze jednou a nemá parametry.
☑ f. Nemá návratový typ.
☐ g. Jmenuje se stejně jako třída.

Úloha **8**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Je dáno:

```
class Data
{
    private:
        int d;
    public:
        void set_d(int d_new)
        {
            d = d_new;
        }

        int get_d() const
        {
            return d;
        }
};
```

Je správné použití příkazu:
`Data data;`

?

Vyberte jednu z nabízených možností:
● Pravda
○ Nepravda

Úloha **9**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Pro třídu *Data*:

```
class Data
{
private:
    int d;

public:
    Data()
    {
        set_d(5);
    }

    Data(int d)
    {
        set_d(2 * d);
    }

    int get_d() const
    {
        return d;
    }

    void set_d(int d)
    {
        this->d = d;
    }
};
```

Je vytvořen objekt *data2* podle následující deklarace:
`Data data2(15);`

Jaká hodnota je poskytnuta výrazem *data2.get_d()* ?

● a. 30
○ b. 10
○ c. 15
○ d. 20

Úloha **10**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Uvažujme třídu *Data*. Jak se nazývá uvedený příkaz?
`Data udaje;`

○ a. Realizace dědičnosti.
● b. Deklarace objektu.
○ c. Definice třídy.
○ d. Uvedený příkaz nelze v objektově-orientovaném programování použít.

Úloha **11**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Je dáno:

```
class Data
{
      private:
            int d;
      public:
            Data(int d)
            {
                  this->d = d;
            }

            void set_d(int d_new)
            {
                  d = d_new;
            }

            int get_d() const
            {
                 return d;
            }
};
```

Je správné použití příkazu:
`Data data;`

?

Vyberte jednu z nabízených možností:
○ Pravda
● Nepravda

Úloha **12**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Je správně uvedený fragment definice třídy?

```
class Data
{
private:
    int d1;
    int d2;

public:
    Data() : Data(5)
    {
        ;
    }

    Data(int d) : Data(d, d)
    {
        ;
    }

    Data(int d1, int d2)
    {
        set_d1(d1);
        set_d2(d2);
    }

    void set_d1(int d1)
    {
        this->d1 = d1;
    }

    void set_d2(int d2)
    {
        this->d2 = d2;
    }

    ... ...
    ... ...
};
```

Vyberte jednu z nabízených možností:
● Pravda
○ Nepravda

Úloha **13**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Agregace je vztah, kdy jeden objekt tvoří celek, který jako části obsahuje jiné objekty.

Vyberte jednu z nabízených možností:
● Pravda
○ Nepravda

Úloha **14**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Je uvedená definice třídy *Data* správná (kompilovatelná)?

```
class Data
{
private:
    int d;

public:
    Data(int d)
    {
        set_d(d);
    }

    int get_d() const
    {
        return d;
    }

private:
    void set_d(int d)
    {
        this->d = d;
    }
};
```

○ a. Ne. Privátní metoda *set_d* je volána v konstrukturou, který je public. Metodu *set_d* nelze takto volat.
● b. Ano.
○ c. Ne. Tato třída nemůže mít takovou privátní metodu.
○ d. Nelze rozhodnout. Záleží na překladači jazyka.

Úloha **15**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Je dána definice třídy:

```
class Data
{
private:
   int d;

public:
   Data(int d)
   {
      set_d(d);
   }

   int get_d() const
   {
      return d;
   }

private:
   void set_d(int d)
   {
      this->d = d;
   }
};
```

Lze pro již vytvořený objekt tento třídy měnit jeho stav?

Vyberte jednu z nabízených možností:
○ Pravda
● Nepravda

Úloha **16**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

V objektově-orientovaném programování, jaké známe principy programování?

○ a. Zapouzdření; Dědičnost. Polymorfizmus nepatří mezi principy programování.
● b. Zapouzdření; Dědičnost;Polymorfizmus
○ c. Zapouzdření. Dědičnost a Polymorfizmus nepatří mezi principy programování.
○ d. Objektový styl programování není postaven na uvedených principech.

Úloha **17**
Hotovo
Bodů 0,00 / 1,00
Úloha s vlaječkou

Pro uvedenou třídu *Data*:

```
class Data
{
private:
    int d;

public:
    Data(int d)
    {
        set_d(d);
    }

    int get_d() const
    {
        return d;
    }

    void set_d(int d)
    {
        this->d = d;
    }
};
```

Je správný zápis vytvoření objektu *data*?
`Data data;`

Vyberte jednu z nabízených možností:
● Pravda
○ Nepravda

Úloha **18**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Uvažujte následující fragment programu. Co bude na standardním výstupu po jeho provedení?

```
int i, &ri = i;
int *pi = &i;
i = 5; ri =10;
*pi = 15;
std::cout << i << " " << ri << std::endl;
```

Odpověď: 15 15

Úloha **19**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Je dán uvedený příkaz.
`Data udaje;`

Vyberte správné tvrzení:

○ a. Objekt *udaje* obsahuje pouze datové položky a žádné metody.
○ b. Třída se jmenuje *udaje* a objekt se jmenuje *Data*.
○ c. Název třídy není uveden. Objekt se jmenuje *udaje*. *Data* je klíčové slovo.
● d. Třída se jmenuje *Data* a objekt se jmenuje *udaje*.

Úloha **20**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Statické datové položky třídy jsou součástí každé instance?

Vyberte jednu z nabízených možností:
○ Pravda
● Nepravda

Úloha **21**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Může být v jazyce C++ zapsán tento fragment kódu?

```
void m()
{
    ;
}

void m(int a, int b)
{
    ;
}
```

Vyberte jednu z nabízených možností:
● Pravda
○ Nepravda

Úloha **22**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Určete chyby v následujícím programu:

```
#include <iostream>
#include <fstream>

int main() {
  ofstream soubor("data.txt");
  soubor << "Vstupní text\n";
  return 0;
}
```

○ a. Je to správně.
○ b. Chyba - soubor nemá určen mód pro zápis "w".
● c. Chyba - soubor není uzavřen

Úloha **23**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Má lambda funkce definované jméno?

Vyberte jednu z nabízených možností:
○ Pravda
● Nepravda

Úloha **24**
Hotovo
Bodů 0,00 / 1,00
Úloha s vlaječkou

Je dána tato funkce/metoda:

```
void m(int a, int b, int c = 4, int d = 5)
{
   ;
}
```

Napište volání metody (příkaz!) takový, aby se pro parametry *a* a *b* použily hodnoty proměnných *c* a *d*, a pro parametr *d* se použila hodnota 10. Nepoužívejte v zápisu mezery!!!

int c = ...;
int d = ...;
...... // volání funkce/metody main

Odpověď: m(c,d,c,10)

Úloha **25**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Jakou hodnotu bude mít symbol EX po následující deklaraci výčtového typu:
`enum {E1,E2=16,E3,E4=E3+5,EX];`

○ a. 22
○ b. 24
● c. 23
○ d. 25

Úloha **26**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Po indikaci chyby přes globální příznak platí:

☑ a. Může nastat více chyb, ale indikovat lze jen poslední.
☑ b. Je nutno chybu hned indikovat.
☐ c. Lze zjistit, kde chyba vznikla.
☑ d. Nelze zjistit, kde chyba vznikla.

Úloha **27**
Hotovo
Bodů 0,00 / 1,00
Úloha s vlaječkou

Může být v jazyce C++ zapsán tento fragment kódu?

```
void m()
{
    ;
}

void m(int a, int b)
{
    ;
}

void m(int a, int b, int c = 4, int d = 5)
{
    ;
}
```

Vyberte jednu z nabízených možností:
● Pravda
○ Nepravda

Úloha **28**
Hotovo
Bodů 1,00 / 1,00
Úloha s vlaječkou

Přátelskou funkci zavedeme pomocí klíčového slova "mate"?

Vyberte jednu z nabízených možností:
○ Pravda
● Nepravda

Úloha **29**
Nezodpovězeno
Počet bodů z 1,00
Úloha s vlaječkou

```
void tisk_znak(char z, char sep)
{
    cout << sep << z << sep;
}
```

Upravte zápis parametrů metody tak, aby se defaultně zobrazovaly znaky z v pomlčkách, např. -a- . Dodržte konvenci zápisu a přepište pouze červeně zvýrazněné.

Např. pro volání *tisk_znak('a')*.

Odpověď:

Úloha **30**
Hotovo
Bodů 0,00 / 1,00
Úloha s vlaječkou

Jak se jmenuje instance standardního výstupního proudu v jazyce C++?

Odpověď: std::cout

Dokončit prohlídku