

## تمرین کلاسی شماره 7 : وراثت و چند ریختی

روش انجام: نوشتن برنامه و ارسال برنامه.

موعد تحویل : چهارشنبه 15 مرداد

### تمرین

1. مزیت‌های استفاده از چندریختی چیست؟ به دو مورد اشاره کنید. می‌توانید از مثال «شکل‌ها» برای توضیح استفاده کنید.
2. دلیل تعریف یک متد به عنوان «مجازی خالص» چیست؟
  - مشخص کنید هر یک از موارد زیر درست هستند یا نه. اگر نه، دلیل آن را توضیح دهید:
  - تمام متدهای مجازی در یک کلاس مجرد باید «مجازی خالص» تعریف شوند.
  - اشاره کردن به یک شیء از نوع یک زیرکلاس با اشاره‌گری با تایپ ابرکلاس خطرناک است.
  - اگر یک کلاس متدی را «مجازی خالص» تعریف کرده باشد، زیرکلاس‌های آن باید آن متد را پیاده‌سازی کنند تا مجرد محسوب نشوند.
3. برای مثال «شکل‌ها»، یک زیرکلاس جدید از Shape تعریف کنید برای پیاده‌سازی مثلث متساوی‌الاضلاع. فرض کنید فیلدهای  $x$  و  $y$ ، مرکز مثلث را مشخص می‌کنند. متدهای لازم را برای آن بازنویسی کنید.
4. به کلاس‌های مثال «شکل‌ها»، متدی برای محاسبه‌ی مساحت اضافه کنید.
5. در نرم‌افزارهای گرافیکی، معمولاً پاره‌خط را هم نوعی شکل محسوب می‌کنند. کلاسی برای این کار از Shape مشتق کنید. فیلدها، متدها و سازنده‌های لازم را برای آن تعریف کنید. اگر مانند سؤال قبل محاسبه‌ی مساحت هم جزء برنامه‌ی شما باشد، چه تغییراتی لازم است در برنامه ایجاد کنیم؟ توجه کنید که مساحت یک پاره‌خط صفر نیست، بلکه تعریف نشده است.
6. در یک شرکت دو نوع کارمند مشغول هستند. نوع اول، حقوق ساعتی دریافت می‌کنند. این دسته، نرخ دستمزد ثابتی برای هر ساعت دارند و حقوق آنها حاصل ضرب این نرخ در تعداد ساعت کار آنها می‌باشد. نوع دوم، حقوق ثابت دارند، به این ترتیب که برای ۱۴۰ ساعت کارکرد خود در ماه، مبلغ ثابتی دریافت می‌کنند و برای بیشتر از ۱۴۰ ساعت، به ازای هر ساعت ۵۰٪ اضافه دریافت می‌کنند. به عنوان مثال، اگر حقوق ثابت کارمندی ۱۴ میلیون ریال باشد، یعنی به ازای هر ساعت صد هزار ریال دریافت می‌کند. حال اگر چنین فردی ۱۸۰ ساعت کار کند، برای ۱۴۰ ساعت اول ۱۴ میلیون ریال و برای ۴۰ ساعت اضافه‌کار، ساعتی صد و پنجاه هزار ریال دریافت می‌کند. در نتیجه کل درآمد این شخص برای ۱۸۰ ساعت بیست میلیون ریال خواهد بود. یک کلاس مجرد کارمند تعریف کنید که دو زیرکلاس برای دو نوع کارمند ساعتی و ثابت داشته باشد. متد `int earnings(int hours)` را برای این کلاس‌ها تعریف کنید. فیلدها، سازنده‌ها و متدهای مورد نیاز را نیز برای این کلاس‌ها تعریف کنید.

7. در ادامه‌ی سؤال قبل، کلاسی به نام **Organization** (سازمان) تعریف کنید که تعداد کارمند (از هر دو نوع) را دارد، و احتمالاً تعدادی زیرسازمان (که خودشان از نوع سازمان هستند). متدی به شکل **int total\_earnings(int avg\_hrs)** برای کلاس سازمان تعریف کنید که مجموع درآمد تمام افراد سازمان (و زیرسازمان‌های آن) را برمی‌گرداند اگر تمام کارمندا **avg\_hrs** ساعت در یک ماه کار کرده باشند.
8. این کلاس‌ها را درنظر بگیرید:

```
class A {
public:
    void f() { cout << "A::f"; }
    virtual void g() { cout << "A::g"; }
    void h() { cout << "A::h"; f(); g(); }
};

class B : public A {
public:
    void f() { cout << "B::f"; }
    void g() { cout << "B::g"; }
    void h() { cout << "B::h"; f(); g(); }
};
```

نتیجه‌ی قطعه کد زیر را (بدون اجرا کردن) مشخص کنید:

```
A a;
B b;
A* ap = &b;
ap->f();
ap->g();
ap->h();
```

9. خطاهای کامپایل برنامه‌ی زیر را (بدون کامپایل کردن) مشخص کنید:

```
#include <iostream>
using namespace std;

class A {
public:
    void f() { cout << "A::f"; }
    virtual void g() = 0;
    void h() { cout << "A::h"; f(); g(); }
private:
```

```
    int a;
    void k() { cout << "A::k"; }
};

class B : public A {
public:
    int f() { cout << "B::f"; return 0; }
    int f(int x) { return x + a; }
    int k() { cout << "B::k"; }
protected:
    void h() { cout << "B::h"; }
};

class C : public A {
public:
    void g() { cout << "C::g"; }
};

int main() {
    A a;
    B b;
    C c;
    A* ap = &c;
    ap->g();
    B* bp = &c;
    bp->g();
}
```