

Point

در این سوال قراره کلاسی که بیانگر یک نقطه (point) می باشد را طراحی کنیم و همچنین یاد بگیریم که چند فایل بنویسیم.

لطفا برای این سوال تعریف کلاس (class definitions) و همچنین function prototype ها را در یک فایل header به نام geometry.h و همچنین تعریف توابع (function definitions) را در فایلی به نام geometry.cpp قرار دهید.

خب همانطور که گفتیم میخواهیم کلاس نقطه را طراحی کنیم. ابتدا کلاسی به نام point بسازید که دارای دو متغیر صحیح خصوصی (private) به نام x و y می باشد.

برای سازنده ی (Constructor) این کلاس یک سازنده در نظر بگیرید که در صورتی که با 0 آرگومان صدا زده شود نقطه ای را مقدار دهی کند که نشان دهنده ی مرکز (0,0) باشد و در صورتی که با دو آرگومان x و y صدا زده شود نقطه ای را ایجاد می کند که در مکان (x,y) واقع شده باشد.

کلاس تون باید شامل member function های زیر باشد.

- تابع Getter که مختصات نقطه در x را بر می گرداند
- تابع Getter که مختصات نقطه در y را بر می گرداند
- تابع Setter که مختصات نقطه در x را set می کند
- تابع Setter که مختصات نقطه در y را set می کند
- تابعی که هر دو مقدار x و y را set می کند
- تابعی که اندازه مختصات را بر میگردداند $\sqrt{x^2 + y^2}$
- تابعی که مختصات نقطه را به صورت (x,y) چاپ می کند.

در نهایت با نوشتن یک main دلخواه کلاس و توابع داخل کلاستون رو تست کنید.

Parray

حال در این سوال می خواهیم کلاسی را طراحی کنیم که بیانگر آرایه از نقاط (Points) هستند. در این سوال قصد داریم که آرایه به صورت داینامیک و پویا سایز خودش را تغییر دهد و سایز خود را ردیابی کند تا اگر بخواهیم آن را به تابعی بفرستیم نیازی به فراهم کردن اندازه ی آرایه نباشد.

کلاسی به نام `PointArray` بسازید که دارای دو عضو `private` می باشد که عبارتند از یک اشاره گر که به ابتدای آرایه ای از نقاط اشاره می کند و همچنین یک عدد صحیح که نشان دهنده ی سایز (اندازه) آرایه می باشد.

کلاستون دارای سازنده (Constructors) های زیر می باشد.

- یک سازنده پیشفرض (default constructor) که آرایه ای با سایز 0 درست می کند.
- یک سازنده که آرایه ای از `point` ها به نام `points` و همچنین یک `int` به نام `size` را به عنوان آرگومان های ورودی می گیرد. و این سازنده باید یک `PointArray` با سایز گفته شده بسازد و مقادیر داخل `points` را کپی کند. شما باید به آرایه ی داخل کلاستون به صورت پویا به اندازه ی گفته شده (size) حافظه تخصیص بدین. (حواستان باشد در جاهایی مثال اینجا که قرار نیست مقداری که به عنوان ورودی به تابع داده می شود را تغییر دهیم حتما آن را به صورت `const` تعریف کنید.)

کلاستون دارای مخرب (Destructor) زیر می باشد

- یک `Destructor` که کارش `Delete` کردن آرایه ی داخلی کلاسمون می باشد.
- از آنجایی که می توانیم بر روی آرایه ی خود اصلاحات انجام دهیم باید راهی پیدا کنیم که آرایه داخلی کلاس را بتونیم بزرگ یا کوچک کنیم.
- یکی از کار هایی که می توانیم انجام دهیم نوشتن یک member function به نام `PointArray::resize(int n)` می باشد که کارش این است که به آرایه ی جدیدی، حافظه ای به سایز `n` تخصیص دهد، مقادیر آرایه ی قبلی را وارد این آرایه ی جدید کند و آرایه ی قبلی را `deallocate` کند.
- در صورتی که سایز آرایه قرار باشد اضافه شود اشکالی ندارد که تابع `resize` فضاهای جدید را مقدار دهی اولیه نکند و پر کردن فضاهای اضافه وظیفه ی کسی می باشد که این تابع را صدا زده.

سپس هر جایی که قرار باشد سائز آرایه را تغییر داده شود (حتی در هنگام clear) از این تابع استفاده می شود. در بعضی موارد لازم است که پس از resize کردن آرایه، آن را به چپ یا راست شیفت بدیم تا جا برای مقادیر جدید باز بشه یا از شر مقادیر قبلی خلاص بشیم :) که البته خیلی کارآمد نمی باشد ولی در این تمرین خیلی نگران کارآمد بودن از این لحاظ نیستیم ولی اگر می خواستین از راه "درست" پیش برین باید حواستان باشد که حداکثر سائز آرایه چقدر است و چقدر از آن پر شده و تنها در صورتی reallocate کنید که به حد فعلی رسیدین و یا زیر یک آستانه ای که در نظر گرفتین قرار گرفتید.

پس تابع `PointArray::resize(int n)` را به عنوان توابع کلاستون اضافه کنید و به آن access modifier مناسبی دهید که تنها توابع داخلی کلاس، بتوانند از آن استفاده کنند.

و در نهایت توابع زیر را به عنوان Member Functions به کلاس خود اضافه کنید.

```
void PointArray::PushBack(const Point &p)
```

یک Point در انتهای آرایه اضافه می کند.

```
void PointArray::insert(const int position, const Point &p)
```

با شیفت دادن آرایه به راست یک point را در جای دلخواه قرار می دهد.

```
void PointArray::remove(const int pos)
```

از جای دلخواه آرایه یک point را حذف می کند.

```
const int PointArray::getSize() const
```

سائز آرایه را باز می گرداند.

```
void PointArray::clear()
```

آرایه را پاک می کند و سائزش را برابر با صفر قرار می دهد.

```
Point *PointArray::get(const int position)
const Point *PointArray::get(const int position) const
```

پوینتری به عنصر موجود در مکان دلخواه که توسط position مشخص شده، باز می گرداند. در صورتی که چنین position ای وجود نداشته باشد null بر می گرداند.

نکات

- از چند فایل به نوشتن استفاده کنید.
- حواستان به چک کردن محدوده آرایه تون در توابع پابلیکتون باشید و در صورت رعایت نکردنش پیام مناسب چاپ کنید.
- با نوشتن main مناسب کلاس و توابع درون کلاس را محک بزنید.

کلاس ماشین

ما میخواهیم کلاسی داشته باشیم که ماشین را توصیف کند. هر ماشین تعدادی چرخ، رنگ بدنه، تاریخ تولید و تعدادی صندلی دارد. رنگ هر ماشین تنها یک رشته است که در آن اسم رنگ ذخیره میشود. برای چرخ ها و صندلی ها نیاز است که کلاس های متناظر با هر کدام از آن ها تعریف شود. برای تاریخ نیز باید کلاسی با نام Date تعریف شود که در آن، تاریخ روز، ماه و سال ذخیره شود. هر کلاسی میتواند توابعی داشته باشد.

برای کلاس تاریخ، توابع زیر را تعریف کنید: تغییر کل تاریخ (که نیاز به هر سه متغیر دارد)

- برگرداندن (return) تاریخ به صورت درست. مثلا 13/12/1397
- برگرداندن (return) تاریخ به صورت متنی. مثلا 13 1397 (یعنی باید بتوانید معادل هر ماه را به کلمه چاپ کنید)

برای کلاس صندلی، رنگ و جنس و تاریخ تولید صندلی را ذخیره کنید و توابع زیر را در آن پیاده سازی کنید:

- برگرداندن (return) هر یک از متغیر های کلاس
- تغییر هر یک از متغیر های کلاس

برای کلاس چرخ، تنها تاریخ تولید آن را ذخیره کنید.

تابع سازنده هر کلاس میتواند اطلاعات ضروری هر کلاس را بگیرد و با آن ها، شی را بسازد. اینکه چه متغیر هایی را در سازنده بگیریم و یا چه مقادیری را توسط توابع دیگر، دست خودتان است.

کلاس ماشین میتواند تعدادی چرخ و صندلی داشته باشد که برای آن ها از کلاس هایی که ساخته اید استفاده کنید. توجه کنید کلاس ماشین را به گونه ای طراحی کنید که بتوانید هر تعداد شی چرخ و صندلی در آن بتوان ذخیره کرد. میتوانید در کلاس ماشین، متغیر هایی برای ذخیره تعداد چرخ ها و صندلی ها بسازید.

توجه پیاده سازی قسمت زیر پیشنهادی بوده و نمره ی اصلی این سوال طراحی درست موارد خواسته شده است

توجه

حال برنامه ای بنویسید که ابتدا از کاربر عددی مبنی بر تعداد ماشین ها بگیرد. سپس به همان تعداد، شی ماشین بسازد و به ترتیب اطلاعات هر ماشین را بگیرد. اطلاعات هر ماشین به شیوه زیر گرفته میشود: (به ترتیب رنگ،

تعداد چرخ، تعداد صندلی. سپس در خط بعد تاریخ تولید ماشین)

blue 4 5
1397 9 21

بعد باید به ازای هر چرخ اطلاعات آن را از کاربر بگیرید و ذخیره کنید(تاریخ تولید)

1396 5 10
1396 5 11
1396 5 10
1396 5 10

و سپس اطلاعات هر صندلی را به ترتیب رنگ، جنس و تاریخ بگیرید

blue plastic 1396 10 10
green plastic 1396 10 10
brown plastic 1397 9 29
brown plastic 1397 10 2
green plastic 1397 10 1

حال اطلاعات هر ماشین را به ترتیب به شیوه زیر پرینت کنید: (این خروجی برای مثال بالا نوشته شده) (قبل از هر ماشین، شماره ماشین را هم پرینت کنید) (ترتیب پرینت اطلاعات چرخ ها و صندلی ها به ترتیبی است که کاربر وارد کرده است)

blue 1397 azar 21
4
1396/5/10
1396/5/11
1396/5/10
1396/5/10
5
plastic 1396/10/10 blue
plastic 1396/10/10 green
plastic 1397/9/29 brown

plastic 1397/10/2 brown

plastic 1397/10/1 green

توجه کنید که در خروجی از هر دو نوع تابع گرفتن تاریخ، استفاده شده است.

توجه کنید: اسامی ماه ها به صورت زیر است:

(farvardin, ordibehesht, khordad, tir, mordad, shahrivar, mehr, aban, azar, deyr, bahman, esfand)

(اختیاری) استرینگ بی استرینگ

در این سوال قصد داریم مفاهیم و توابع مربوط به مدیریت رشته را به همراه مفهوم کلاس پیاده سازی نماییم :

بدین منظور class با اسم String را با شرایط زیر پیاده سازی نمایید:

- بخش private شامل دو قسمت زیر است: یک متغیر از جنس char* به اسم str و یک متغیر از نوع int به اسم size
- بخش public نیز شامل توابع زیر است
- یک تابع سازنده که بدون دریافت ورودی تمام attribute ها را به صورت دلخواه و مناسب مقداردهی اولیه میکند.
- یک تابع سازنده که با دریافت ۲ ورودی مناسب attribute ها را ست می کند.
- یک تابع سازنده که با دریافت یک ورودی از جنس char[] مقدار str را نیز ست میکند (متغیر size نیز متناسباً باید مقداردهی شود).
- یک تابع مخرب بنویسید.

در فاز بعدی توابع زیر را به صورت توصیف شده برای قسمت public پیاده سازی نمایید:

```
void setStr(char*)
```

این تابع با دریافت اشاره گری به کاراکترها، مقدار str را تغییر میدهد.

دقت شود که در این حالت مقدار size نیز باید از طریق تابع setSize به روز رسانی شود.

```
char * getStr()
```

این تابع مقدار موجود در متغیر str را بر میگرداند.

```
void setSize(int)
```

وظیفه این تابع ست کردن مقدار size می باشد.


```
int getSize()
```

این تابع مقدار size را بر میگرداند.

```
char* subStr(int index1, int index2)
```

این تابع زیر رشته ای که از اندیس index1 آغاز و به index2 ختم می شود را بر میگرداند.

```
void append(char* temp)
```

این تابع رشته temp را به انتهای رشته ی str اضافه کرده و در str ذخیره می کند.

```
int findSubStr(char* temp)
```

خروجی این تابع در صورتی که رشته str دارای زیررشته ی temp باشد برابر اولین اندیس تطابق دو رشته str و temp و در غیر این صورت برابر -۱ است.

```
bool strCmp(char* temp)
```

خروجی ین تابع در صورتی که دو رشته str و temp کاملاً با هم برابر باشند، true و در غیر این صورت false است.

- نکته ۱) برای برنامه نویسی این سوال استفاده از کتابخانه های stdlib و string مجاز نمی باشد.
- نکته ۲) برای برنامه خود رابط کاربری مناسب طراحی کنید.