مدرسان: **رامتین خسروی، محمدامین صادقی**
طراحان: **هادی صفری، محمدمهدی مهدی‌زاده**

مهلت تحویل: جمعه ۱۷ آذر ۱۳۹۶، ساعت ۲۳:۵۵

ربات باب راس (Bob Ross Bot)

ربات نقاش

آلیس قصد دارد برای گرامی‌داشت بیست‌ودومین سالگرد درگذشت نقاش و معلم فقید، باب راس، از یک ربات نقاش رونمایی کند. از آن‌جا که او آدمی عجول و وسواسی است و البته عمرش را با رنگ روغن گذرانده است نه زیر نور نمایشگرها، کمی زودتر دست به کار شده است و از دستیاران آموزشی درس برنامه‌نویسی پیشرفته خواسته است کمکش کنند؛ آن ها نیز طی جلسه‌ای مهم تصمیم گرفتند این مسئولیت خطیر را به دانشجویان واگذار کنند.

ساختار ورودی

ربات باب راس ــ که از این پس آن را به‌اختصار ربات می‌نامیم ــ دستورهای کاربر را که به زبانی ساده هستند می‌گیرد (آلیس معتقد است این کلمات را باب در خواب به او گفته است و اصرار دارد ربات عیناً همین دستورها را بفهمد) و بر اساس آن‌ها یک فایل گرافیکی برداری مطابق استاندارد SVG می سازد. دستورهای کاربر در ادامه بیان خواهند شد.

چارلز که از علاقه‌مندان هنر مرحوم باب راس است اصرار دارد از ربات در طراحی وب‌سایتش کمک بگیرد. چارلز سلایق عجیبی دارد؛ مثلاً می‌خواهد به‌جای فرمت SVG از تصاویر Canvas در طراحی هایش استفاده کند. بعد از ساعت ها مذاکره، دستیاران آموزشی توانستند چارلز را راضی کنند تصاویر Canvas فقط بعضی از ویژگی‌های خاص را ــ که در ادامه مشخص شده‌اند ــ داشته باشد. طی مذاکرات بعدی که با حضور استادان درس صورت گرفت، چارلز رضایت داد که تولید خروجی Canvas اختیاری باشد. اما او در عوض خواست تا ربات توانایی ذخیره‌کردن و بارگیری فایل‌هایی با پسوند bob را هم داشته باشد.

دستورهای ورودی ــ خروجی

کار با فایل‌های bob

همان‌طور که بیان شد، برنامه‌ی شما باید بتواند با فایل‌هایی با پسوند bob کار کند. این فایل‌ها فایل‌هایی متنی‌اند که دستورهایی مشابه دستورهایی که آلیس درخواست کرده بود دارند. در ابتدای شروع به کار برنامه باید بتوان یک فایل bob را بارگیری کرد و در ادامه از طریق ورودی استاندارد به برنامه دستورهای جدید داد، یا یک صفحه‌ی جدید ساخت و همه‌ی دستورها را مستقیماً از ورودی استاندارد وارد کرد.

در هر نقطه‌ای از اجرای برنامه با دستور save باید بتوان محیط کار فعلی را در یک فایل bob ذخیره کرد.

```
save <file_address>
```

تولید فایل‌های نهایی

در هر جای برنامه با دستور export برنامه باید خروجی‌های مورد انتظار را تولید کند. با توجه به پسوند فایل نوع خروجی تعیین می‌گردد. از پسوند svg برای فرمت svg و از پسوند html یا htm برای فرمت Canavas استفاده می‌شود. اگر پسوند دیگری ــ از جمله پسوند bob ــ استفاده شده بود باید پیام خطای مناسبی تولید شود. (اگر بخش Canvas را پیاده‌سازی نکرده‌اید، با پسوندهای html و htm نیز همانند پسوندهای ناشناس برخورد کنید.)

```
export <file_address>
```

ابعاد صفحه

با دستور init که اولین دستور هر صفحه است و پارامتری از نوع مختصات دارد ابعاد صفحه تعیین می‌شود. ویژگی‌های مختصات در ادامه تشریح خواهد شد.

```
init <dimension>
```

این دستور همواره باید اولین دستور ورودی استاندارد ــ وقتی فایلی بارگیری نمی‌شود ــ و اولین دستور هر فایل bob باشد. استفاده از این دستور در ادامه‌ی برنامه مجاز نیست و نمی‌توان در میانه‌ی کار ابعاد صفحه را تغییر داد.

```
init <640,480>
```

برخی قواعد کلی و دستورهای عمومی

نام‌گذاری‌ها

در هر نوع نام‌گذاری، نام مجاز از حروف بزرگ و کوچک انگلیسی، اعداد و زیرخط(_) تشکیل شده است. کلمات کلیدی مانند نام اشیا، عبارت with و کلمات مشابه نام‌گذاری معتبری نیستند. نام ویژگی‌ها کلمه‌ی کلیدی محسوب نمی‌شود. می‌توانید فرض کنید در زمان اجرا نام نامعتبری به برنامه داده نمی‌شود.

رنگ[1]

در همه‌ی دستورها برای توصیف رنگ‌ها از روش یکسانی استفاده می‌شود که در این بخش توضیح داده می‌شود. یک رنگ را می‌توان به چند شکل توصیف کرد:

- نام رنگ: نام هر یک از رنگ‌های استاندارد HTML می‌تواند به‌عنوان رنگ استفاده شود. این رنگ‌ها را می‌توانید مثلاً در سایت w3schools مشاهده کنید.[2]
- کد رنگ: کد رنگ به‌شکل یک عدد ۶رقمی هگزادسیمال که بعد از نویسه‌ی # می‌آید می‌تواند برای توصیف یک رنگ به کار رود.

مختصات[3]

در تمامی دستورهای آلیس برای بیان هر نوع مختصات (مکان، فاصله، اندازه و ...) از روش واحدی تبعیت می‌شود. هر نقطه روی صفحه‌ی مختصات از زوج مرتبی شامل دو مختصه‌ی اعشاری *مختصه‌ی در راستای محور x* و *مختصه‌ی در راستای محور y* تشکیل می‌شود که با کاما (,) از هم جدا می‌شوند. در دو طرف زوج مرتب از <> استفاده می‌شود.

---

[1] Color

[2] https://www.w3schools.com/tags/ref_colornames.asp

[3] Coordination

```
draw circle with center=<5,3.5>, radius=6
```

مختصات در دستگاه مختصات دکارتی با جهت محورهای رایج در برنامه‌نویسی بیان می‌شوند. در این دستگاه، نقطه‌ی بالا ـ چپ مبدأ مختصات است.

همهٔ مختصات مربوط به مکان ها نسبت به نقطه‌ی بالا ـ چپِ دربرگیرنده‌ی بلافصلِ شیء یا گروه فعلی بیان می شوند. این دربرگیرنده ممکن است یک گروه یا ریشه‌ی فایل باشد. درباره‌ی گروه‌ها در ادامه به‌تفصیل سخن گفته خواهد شد.

در هنگام ساخت اشیا و ... ، آن‌ها عضو هیچ گروهی نیستند؛ بنابراین مکان‌ها نسبت کل مبدأ مختصات صفحه بیان می‌شوند.

توجه کنید که مختصات منفی هنگام بیان مکان‌های نسبی مقادیر معتبری هستند.

آرایه

هنگام نمایش یک آرایه یا لیست در دستورها، درایه‌ها با کاما (,) از هم جدا می‌شوند و در دو طرف لیست نیز نویسه‌های [] قرار می‌گیرند.

```
draw polygon with points=[<1,2>,<3,4>,<5,6>]
```

تغییر ترتیب قرارگیری

همه‌ی موجودیت‌ها بر اساس ترتیب اضافه‌شدنشان از زیر به رو چیده می‌شوند. برای تغییر این ترتیب ـ صرفاً بین هم‌رده‌ها ـ از دستورهای `bring up` و `bring down` استفاده می‌شود.

```
bring (up | down) <name>
```

اگر یک موجودیت بین هم‌رده‌هایش بالاترین یا پایین‌ترین بود، به‌ترتیب دستورهای `bring up` و `bring down` روی آن تأثیری نخواهد داشت و پیغام خطایی تولید نمی‌شود.

```
bring up g1
bring down t3
```

اشیا[4]

آلیس انتظار دارد باب بتواند با گرفتن دستور مناسب، اشیائی را بسازد. هر شیء تعدادی *ویژگی* (خصیصه) دارد که ممکن است در زمان ساخت شیء یا بعد از آن به شیء داده شوند. جزئیات ویژگی‌ها در ادامه بیان خواهد شد.

برای ساخت یک شیء از دستور `draw` استفاده می‌شود:

```
draw <shape_type> <shape_name> with [<attr>=<val>[, <attr>=<val>]]
```

هر شیء یک نام (name) دارد که ممکن است هنگام ساخت آن صراحتاً ذکر شود یا نشود.

- اگر هنگام ساخت یک شیء نامی داده شود که قبلاً وجود داشته است، باید پیغام خطای مناسبی نمایش داده شود و شیء ساخته نشود.
- اگر هنگام ساخت یک شیء نام آن ذکر نشود، بر اساس نوع شیء نامی به شکل `<shape_type>i` به آن اختصاص می‌یابد؛ مانند: `circle0`. `i` عددی صحیح است که از ۰ آغاز می‌شود و هر بار یکی زیاد می‌شود. اگر نام پیش‌فرض تکراری بود، از عدد بعدی استفاده می‌شود.

[4] Shape

هر کدام از اشیا تعدادی ویژگی اجباری دارند و ممکن است تعدادی ویژگی اختیاری نیز داشته باشند. برخی از ویژگی‌ها بین همه‌ی اشیا مشترکند و برخی دیگر فقط به انواع خاصی از اشیا اختصاص دارند. درباره‌ی ویژگی‌ها در ادامه توضیحات بیشتری ارائه خواهد شد.

برای نابودی یک شیء می‌توان از دستور erase استفاده کرد.

```
erase <shape_name>
```

دایره

دایره با عنوان circle شناخته می‌شود و حتماً باید ویژگی‌های center (مختصات) و radius (طول) را داشته باشد.

```
draw circle c1 with center=<3,2>, radius=1
draw circle c2 with center=<5,7>, radius=3, color=blue, opacity=0.5, border=<black,1>
```

مستطیل

مستطیل با عنوان rectangle شناخته می‌شود و حتماً باید ویژگی points (آرایه‌ای دوتایی از مختصات) را که نشانگر دو رأس مقابل مستطیل است داشته باشد.

```
draw rectangle r1 with points=[<0,0>,<3,6>]
```

خط

خط با نام line شناخته می‌شود و حتماً باید ویژگی points (آرایه‌ای دوتایی از مختصات) را که نشانگر دو سر خط است داشته باشد.

```
draw line l1 with points=[<0,0>,<3,6>]
```

متن

متن ــ به‌عنوان نوعی شیء ــ با نام script شناخته می‌شود. هر متن یک ویژگی text (رشته) دارد؛ اما ممکن است این ویژگی هنگام تعریف متن ذکر نشود. در این حالت، ویژگی text متن را برابر رشته‌ای خالی در نظر بگیرید. محل قرارگیری متن‌ها با ویژگی اجباری position (مختصات) تعریف می‌شود.

```
draw script t1 with text="R.i.P dear Bob!", position=<0,0>
```

چندضلعی (اختیاری)

چندضلعی با نام polygon شناخته می‌شود و حتماً باید ویژگی points (آرایه‌ای از مختصات) را داشته باشد.

```
draw polygon p1 with points=[<0,0>,<3,6>,<1,4>]
```

بیضی (اختیاری)

بیضی با عنوان ellipse شناخته می‌شود و حتماً باید ویژگی‌های center (مختصات) و radius (مختصات) را داشته باشد.

```
draw ellipse e1 with center=<3,2> radius=<1,1.5>
```

عکس (اختیاری)

عکس با نام `picture` شاخته می‌شود و حتماً باید ویژگی `address` داشته باشد. عکس باید به‌شکل داینامیک به خروجی نرم‌افزار وابسته باشد و هنگام مشاهده‌ی فایل خروجی، ابزار مشاهده ـ مثلاً مرورگر وب ـ آن را بارگیری کند. آدرس عکس به صورتی نسبی به ربات داده می‌شود. برای مشخص‌کردن محل و ابعاد عکس همانند مستطیل عمل می‌شود.

```
draw picture with address="bob.png", points=[<0,0>,<3,6>]
```

گروه‌ها[5]

آلیس بسیار وسواسی است و به نظم و ترتیب اهمیت زیادی می‌دهد؛ به همین دلیل اصرار دارد گروه‌هایی برای دسته‌بندی اشیا وجود داشته باشد. هر گروه می‌تواند شامل تعدادی شیء و گروه دیگر باشد. از دستور `group` برای ساخت یک گروه جدید استفاده می‌شود.

```
group <group_name> with [<attr>=<val>[, <attr>=<val>]]
```

قوانین نام‌گذاری اشیا برای گروه‌ها هم برقرار است. نام‌گذاری پیش‌فرض گروه‌ها با پیشوند `group` صورت می‌گیرد؛ مانند: `group0`.

هر گروه حتماً ویژگی `inherit` را دارد که می‌تواند یکی از مقادیر `default` و `force` را داشته باشد. این ویژگی نحوه‌ی به‌ارث‌رسیدن ویژگی‌های گروه به اعضای آن را نشان می‌دهد. اگر این ویژگی صراحتاً هنگام ساخت گروه ذکر نشود، مقدار آن برابر `default` در نظر گرفته می‌شود. ویژگی `inherit` بر روی تمام اعضای مستقیم و غیرمستقیم (باواسطه) گروه تأثیر می‌گذارد. لازم به ذکر است که ویژگی `position` هیچ‌گاه به ارث نمی‌رسد.

- مقدار `default`: اعضای گروه صرفاً اگر برخی از ویژگی‌های گروه را نداشته باشند آن‌ها را به ارث می‌برند. مقادیر تعریف‌شده بازنویسی نمی‌شوند.
- مقدار `force`: اعضای گروه بدون آن‌که از داشتن یا نداشتن یک ویژگی تأثیر بپذیرند همه‌ی ویژگی‌های گروه را به ارث می‌برند. مقادیر تعریف‌شده بازنویسی می‌شوند.

هر گروه حتماً باید ویژگی `position` (مختصات) را داشته باشد که نشان‌گر مختصات نقطه‌ی بالا ـ چپ گروه است. مختصات تمام «مکان»های اعضای مستقیم گروه نسبت به این نقطه محاسبه می‌شود؛ در نتیجه با جابه‌جایی یک گروه تمام اعضای آن هم جابه‌جا می‌شوند. با توجه به این‌که در دنیای واقعی جابه‌جایی بسیار بیشتر از افزودن به گروه یا حذف از آن و خروجی‌گرفتن اتفاق می‌افتد، انتظار می‌رود نسبی‌بودن مختصات مکان‌ها در پیاده‌سازی شما هم در نظر گرفته شود.

```
group g1 with inherit=default, position=<5,4>, color=#eeeeee
group g2 with inherit=force, position=<5,2>
```

پاک‌کردن

از دستور `ungroup` برای نابودی یک گروه استفاده می‌شود. با نابودی یک گروه اعضای آن به عضویت دربرگیرنده‌ی بلافصلِ گروهِ نابودشده درمی‌آیند؛ مثلاً اگر `t1` عضو گروه `g1`، `g1` عضو گروه `g2` و `g2` عضو `g3` بوده باشند، با نابودی `g1`، شیء `t1` به عضویت گروه `g2` درمی‌آید. بعد از نابودی گروهی که دربرگیرنده‌ای نداشته باشد، اعضای مستقیمش عضو هیچ گروهی نخواهند بود.

```
ungroup <group_name>
```

[5] Group

پاک‌شدن یک گروه نباید در ترتیب قرارگیری اعضای گروه نابودشده نسبت به اعضای دربرگیرنده‌ی گروه نابودشده تغییری ایجاد کند.

```
ungroup g1
```

انتقال اعضا و اضافه‌کردن آن‌ها به گروه‌ها

برای اضافه‌کردن یک عضو به اعضای گروه از دستور move استفاده می‌شود.

```
move <member_name> to [<group_name>]
```

در این حالت اگر عضو مذکور قبلاً عضو گروه دیگری بوده است عضویتش لغو می‌شود و در هر حال به عضویت گروه ذکرشده درمی‌آید. اگر نام گروه در این دستور نیاید، عضویت عضو ذکرشده در همه‌ی گروه‌ها لغو می‌شود و به‌عنوان یک موجودیت مستقل بدون هیچ دربرگیرنده‌ای در صفحه قرار می‌گیرد.

توجه کنید که با تغییر عضویت یک عضو نباید مختصات ظاهری آن تغییری بکند؛ پس ممکن است نیاز باشد مقادیر ذخیره‌شده برای «مکان»های آن عضو تغییر کنند.

با موجودیت‌هایی که به یک گروه اضافه می‌شوند از نظر ترتیب قرارگیری مانند موجودیتی که تازه ساخته شده است برخورد می‌شود.

```
move t1 to g1
move t2
```

مشاهده‌ی لیست اعضا

برای دیدن لیست اعضای یک گروه از دستور list استفاده می‌شود.

```
list [<group_name>]
```

اگر در این دستور نام گروه نیاید، لیست موجودیت‌هایی که عضو هیچ گروهی نیستند نمایش داده می‌شوند.
این لیست باید به ترتیب قرارگیری اشیا ــ از زیر به رو ــ چاپ شود.

```
list g1
list
```

ویژگی ها[6]

همان‌طور که قبلاً به‌اختصار بیان شد، هر شیء یا گروه ممکن است تعدادی ویژگی (خصیصه) داشته باشد. بعضی از ویژگی‌ها بین همه‌ی اشیا و گروه‌ها مشترکند و برخی فقط برای انواع خاصی از اشیا یا گروه‌ها تعریف می‌شوند.

ویژگی‌ها به شکل `val=attr` در دستورهای مختلف استفاده می‌شوند که `val` نام یک ویژگی و `attr` مقدار آن است. هنگامی که در دستوری چند ویژگی استفاده می‌شود، ویژگی‌های مختلف با نویسه‌های کاما و فاصله (,) از هم جدا می‌شوند.

ویژگی	توضیح	نوع	دامنه
`center`	مرکز شکل	مختصات	دایره و بیضی
`radius`	شعاع	عدد اعشاری نامنفی	دایره
`radius`	شعاع (در راستای محورهای x و y)	مختصات	بیضی
`points`	نقاط شکستگی	آرایه از مختصات	مستطیل، چندضلعی، خط، عکس
`font`	نحوه‌ی نمایش متن	`<string font_name (without whitespace),int size>`	متن
`color`	رنگ پرکننده‌ی شکل	رنگ	همه‌ی اشیا، گروه
`border`	رنگ و ضخامت خط دربرگیرنده‌ی شکل از خطوط ساده برای دور اشکال استفاده کنید.	`<color,float width>`	همه‌ی اشیا، گروه
`address`	آدرس	آدرس‌دهی معمول سیستم‌عامل یونیکس و صفحات وب به‌شکل رشته (احاطه‌شده با دو نویسه‌ی ")	عکس
`text`	متن	رشته (احاطه‌شده با دو نویسه‌ی ")	متن
`opacity`	شفافیت (۰ به‌معنی ناپدیدشدن و ۱۰۰ به‌معنی نمایش عادی است.)	درصد (به‌شکل عدد اعشاری نامنفی)	همه‌ی اشیا، گروه
`rotate`	چرخش ساعت‌گرد	درجه (به‌شکل عدد اعشاری)	همه‌ی اشیا، گروه
`inherit`	نحوه‌ی به‌ارث‌رسیدن ویژگی‌ها به اعضا	نحوه‌ی ارث‌بری	گروه
`position`	مختصات نقطه‌ی بالا_چپ	مختصات	متن، گروه

تغییر ویژگی‌ها

بعد از ساخت هر شیء یا گروه باید بتوان با استفاده از دستور touch و نام شیء یا گروه، مقدار ویژگی‌های یک شیء یا گروه خاص را تغییر داد یا برای یک ویژگی که تاکنون مقداری نداشته است مقداری تعیین کرد.

```
touch <name> <attr>=<val>
```

برای مثال، برای تغییر آدرس یک تصویر به نام p1 از دستور زیر استفاده می‌شود:

```
touch p1 address="Bob.svg"
```

[6] Attribute

نحوه‌ی تحویل

فایل برنامه‌ی خود را با نام A6-SID.zip در صفحه‌ی CECM درس بارگذاری کنید که SID شماره‌ی دانشجویی شماست؛ برای مثال، اگر شماره‌ی دانشجویی شما ۸۱۰۱۱۲۳۴۵ است، نام فایل شما باید A6-810112345.zip باشد.

لطفاً از روش‌های دیگر فشرده‌سازی مانند rar یا tar.gz استفاده نکنید.

هدف این پروژه آشنایی شما با وراثت و چندریختی و تعمیق آموخته‌هایتان درباره‌ی طراحی شیءگرا است؛ بنابراین طراحی مناسب پروژه از اهمیت بالایی برخوردار است. **شما باید دیاگرام کلاس‌هایتان را هنگام تحویل حضوری همراه خود بیاورید.** همچنین شما باید بتوانید در زمان تحویل دستیاران آموزشی را قانع کنید طراحی مناسبی داشته‌اید و تصمیم‌هایتان بر اساس دلایل معقول اتخاذ شده است. از شما خواسته خواهد شد که نحوه‌ی اضافه‌کردن یک نوع ویژگی جدید (مانند سایه)، یک نوع شیء جدید (مانند ستاره) و یک نوع خروجی جدید به برنامه (مانند VML) را بر اساس طراحی خود توضیح دهید. طراحی شما باید به گونه‌ای باشد که اضافه‌کردن این ویژگی‌ها به طراحی، آن را دچار تغییرات اساسی نکند.

- برنامه‌ی شما باید در سیستم‌عامل لینوکس و با مترجم g++ با استاندارد c++98 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود. همچنین، برنامه‌ی شما باید Makefile داشته باشد.
- به فرمت و نام فایل‌های خود دقت کنید.
- از صحت فرمت ورودی‌ها و خروجی‌های برنامه‌ی خود مطمئن شوید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.
- مطالب بیان‌شده در تالار گفت‌وگوی پروژه در سامانه‌ی CECM جزء صورت پروژه حساب می‌شوند.