



دانشکده ریاضی و علوم کامپیوتر

پروژه میانترم برنامه سازی پیشرفته

Class of Life

استاد: ابراهیم اردشیر لاریجانی

توضیحات اولیه

نوکلئوتید: مولکول های سازنده RNA و DNA هستند که به صورت G و C و T و A نمایش داده می شوند.

مکمل نوکلئوتید: هر نوکلئوتید مکملی دارد که با آن پیوند شیمیایی تشکیل میدهد:
(A با T) و (C با G) مکمل یکدیگرند.

RNA: یک توالی یک رشته ای از نوکلئوتیدها.

مانند:

AATCCGATCCGCCTTA ...

DNA: یک توالی دو رشته ای از نوکلئوتیدها که رشته دوم مکمل رشته اول است.

مانند:

AATCCGATCCGCCTTA ...

TTAGGCTAGGCGGAAT...

کروموزوم: شی ای در سلول که فقط شامل یک DNA است.

برنامه اول

متغیرها و متودهای کلاس Gene

- متغیرهای RNA و DNA.
 - متودی برای دریافت محتوای ژنی (DNA و RNA).
 - متودی برای ساخت DNA از روی RNA. (DNA ساخته شده از روی RNA را خروجی دهد) (راهنمایی: باید رشته مکمل تولید شود)
 - متودی برای ایجاد جهش کوچک در DNA و RNA.
- مثال:

Method_name(A, C, n)

بجای A، C را n بار جایگزین کن.

توجه: در DNA که دو رشته‌ای است، بعد از انجام جهش، در رشته دوم هم باید جفت صحیح آن جایگزین شود.

RNA: ATCGATCG

Method_name(A, C, 2) CTCGCTCG

DNA: ACGGCCTAA

TGCCGGATT

Method_name(A, C, 2) CCGGCCGAA

GGCCGGCTT

- متودی برای ایجاد جهش بزرگ در DNA و RNA.
مثال:

Method_name(S_1, S_2) $|S_1|, |S_2| \geq 0$

به اولین زیررشته S_1 که رسید، زیررشته S_2 را جایگزین آن کند.

توجه: در DNA که دو رشته‌ای است، در رشته دوم باید مکمل S_2 جایگزین شود.

DNA: AAGTCTCAGT
TTCAGAGTCA

Method_name(TCAG, ATC) ATAGTCTCAGT
→ TATCAGTCA

- متودی برای ایجاد جهش معکوس.
مثال:

Method_name(S_1) $|S_1| \geq 0$

به S_1 که رسیدی، آن را در رشته معکوس کن.

توجه: در DNA که دو رشته‌ای است، در رشته دوم باید مکمل S_1 معکوس شود.

توجه: تمام متودهای بالا باید بر روی هر دو متغیر RNA و DNA اعمال شوند.

متغیرها و متودهای کلاس Cell

- تعداد کروموزوم‌ها را (هر کروموزوم شی‌ای از Gene است) ذخیره و محتوای ژنی آنها را دریافت کند. (کروموزوم فاقد RNA است)
- متودی برای مرگ سلولی.
- اگر در کروموزومی، در بیش از 5 نقطه از DNA، پیوندی تشکیل نشده بود، یا تعداد جفت A و T بیش از 3 برابر جفت C و G بود، شی را حذف کن.
- متود جهش بزرگ را بازنویسی کنید. (همنام با متود جهش بزرگ در کلاس Gene)

مثال:

Method_name(S_1 , n , S_2 , m)

اولین زیررشته S_1 از کروموزوم n ام را با اولین زیررشته S_2 از کروموزوم m ام عوض کن.

n ام	AC	S_1	AT
	TG		TA

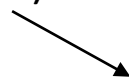
n ام	AC	S_2	AT
	TG		TA

m ام	CG	S_2	AC
	GC		TG

m ام	CG	S_1	AC
	GC		TG

- متود جهش کوچک را بازنویسی کنید.

Method_name(A, C, n , m)



شماره کروموزوم

- متود جهش معکوس را بازنویسی کنید.

Method_name(S_1 , n)

شماره کروموزوم

- متودی برای پیدا کردن پالیندروم مکمل به طول بیشتر از 2 در کروموزوم شماره n و چاپ کردن آن.

رشته‌ای را پالیندروم می‌گوییم هرگاه به این صورت باشد:

$\overline{WW^R}$

به مثال‌های زیر توجه کنید:

ATTCGAAT , AATT , CGCG , ...

این برنامه را در فایل‌های Cell.cpp و Cell.h ذخیره کنید.

برنامه دوم

متغیرها و متدهای کلاس Animal

- متدی برای محاسبه درصد تشابه ژنتیکی.
محتوای ژنتیکی کروموزوم‌های دو جاندار را مقایسه کنید و درصد تشابهی را خروجی دهید.
نکته: می‌توانید از الگوریتم‌های ابتکاری و یا الگوریتم‌های موجود استفاده کرده و طبق الگوریتم استفاده شده به شما نمره امتیازی هم تعلق خواهد گرفت.
- بازنویسی اپراتور == برای تشخیص یک گونه بودن.
اگر محتوای ژنتیکی دو Animal بیش از 70 درصد شباهت داشت و تعداد کروموزوم‌ها برابر بود، هردو از یک گونه هستند، پس در نتیجه true بر می‌گرداند.
- متدی برای تولید مثل غیرجنسی.
این متد تعداد کروموزوم‌ها را دو برابر می‌کند (2n) و نیمی از آنها (n) را به صورت رندوم انتخاب کرده و سپس شی‌ای دیگر برمی‌گرداند.
- بازنویسی اپراتور + برای تولید مثل جنسی.

obj1 + obj2

obj1 و obj2 ابتدا هر کدام تولید مثل غیر جنسی انجام می دهند و در نتیجه دو شی هر کدام با n کروموزوم پدید می آیند. سپس نصف کروموزوم ها از شی اول و نصف از شی دوم به طور رندوم انتخاب می شوند. اپراتور باید بچه آنها را خروجی دهد.

توجه: اگر n فرد بود تولید مثل جنسی انجام نمی شود.

متغیرها و متدهای کلاس Virus

(امتیازی)

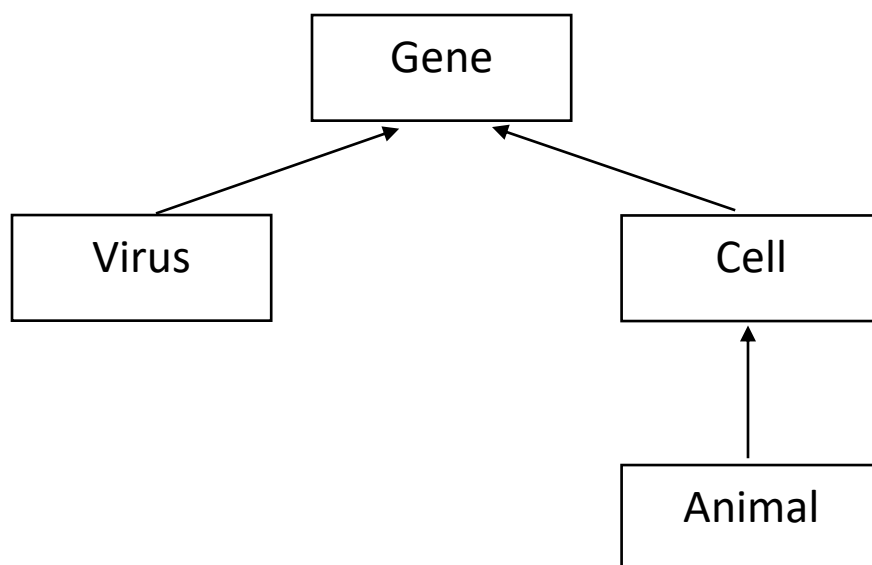
- ویروس‌هایی که در این کلاس می‌سازیم فقط حاوی RNA هستند.
 - کلاس Virus دوست (friend) کلاس Animal است.
 - متدی برای تشخیص مضر بودن ویروس.
- زمانی ویروس باعث بیماری در یک Animal می‌شود که بزرگ‌ترین زیررشته مشترک کروموزوم‌های Animal، خودش یا مکملش یک زیررشته از RNA ویروس باشد.

این برنامه را در فایل‌های
Animal.cpp و Animal.h ذخیره
کنید.

نکات کلیدی

1. برنامه‌ی شما باید user-friendly باشد. یعنی شامل منویی باشد که تمام متدهای بالا را بتوان اجرا کرد (ساخت اشیاء، اعمال جهش و ...) و بعد از اعمال هر متدی اثر آنرا بتوان مشاهده کرد.
2. پروژه باید در قالب تیم‌های دو نفره انجام شود.
3. استفاده از Git و GitHub اجباری ست. برای دوستانی که با آنها آشنایی ندارند کلاس برگزار خواهد شد.
4. ترتیب ارث‌بری ها که در نمودار نشان داده شده را نیز رعایت کنید.

OOP inheritance diagram



موفق باشید